**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**

**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**

**Université Mohamed Khider – BISKRA**

**Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie**

**Département d'informatique**

**N° d'ordre : IVA19/M2/2019**

**Mémoire**

Présenté pour obtenir le diplôme de master académique en **Informatique**

Parcours : **IVA**

---

# Un controleur PID pour la stabilisation de la temperature dans une machine industrielle

---

**Par :**

**KAHLOUL MOHAMED LAMINE**

Soutenu le 11juillet 2019, devant le jury composé de :

| | | |
|---|---|---|
| Djerou leila | Dr | Président |
| Ababsa Tarek | Dr | Rapporteur |
| Chighob Faouzia | MAA | Examinateur |

# DÉDICACES

**Je dédie ce modeste travail :**

**A toute ma famille**

**A tous mes amis**

**A tous ceux qui m'ont aidé**

# *Remerciements*

*Je tiens à exprimer toute ma reconnaissance à ma directrice de mémoire, Monsieur Tarek abbabsa. Je la remercie de m'avoir encadré, orienté, aidé et conseillé.*

*J'adresse mes sincères remerciements à tous les professeurs, intervenants et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé mes réflexions et ont accepté de me rencontrer et de répondre à mes questions durant mes recherches.*

*Je remercie mes très chers parents, qui ont toujours été là pour moi. Je remercie mes sœurs pour touts, et mon frère, pour leurs encouragements.*

*Enfin, je remercie mes amis et grand remercie pour Khaled Troudi qui ont toujours été là pour moi. Leur soutien inconditionnel et leurs encouragements ont été d'une grande aide.*

*À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude*

**Sommaire**

Chapitre I. Généralités sur les systèmes asservis ..... 2

I.1. Introduction ..... 2

I.2. Notion de système asservi ..... 2

I.3. Chaîne de régulation automatique ..... 3

I.3.1. Système en boucle ouverte ..... 3

I.3.2. Système en boucle fermée ..... 4

I.4. Modélisation ..... 6

I.4.1. Schéma physique ..... 6

I.4.2. Schéma fonctionnel ..... 6

I.4.3. Modèle mathématique : ..... 7

I.5. Caractéristiques dynamiques d'un système asservi ..... 7

I.5.1. Performances d'un système asservi ..... 7

I.6. Contrôles automatiques ..... 10

I.6.1. PID Controller ..... 10

I.6.2. On/Off Controller ..... 12

I.6.3. Réseau de neurones ..... 13

I.7. Conclusion ..... 17

Chapitre II. PID ..... 19

II.1. Introduction ..... 19

II.2. Comment Travailler ..... 19

II.3. Les Régulateurs ..... 20

II.3.1. Les Régulateurs simples ..... 20

II.3.2. Les Régulateurs complexes ..... 27

II.4. Conclusion ..... 35

Chapitre III. Matérielle et logicielle ..... 37

III.1. Introduction ..... 37

III.2. Partie matérielle ..... 37

III.2.1. La carte ArduinoUno ..... 37

III.2.2. Le Capteur ..... 49

III.2.3. Le Relais ..... 49

III.3. Programmation en arduino ..... 49

III.3.1. Définition d'un programme arduino ..... 49

III.3.2. La structure d'un programme ..... 49

III.3.3. Coloration syntaxique ..... 50

III.3.4. La syntaxe du langage : ..... 51

III.4. ..... 54

III.5. Conclusion : ..... 57

Chapitre IV. L'implémentation ..... 59

IV.1. Introduction ..... 59

Sommaire

# Table des figures :

**Figure I-1 : Schéma bloc élémentaire..... 2**

**Figure I-2: Schéma fonctionnel d'un système asservi ..... 3**

**Figure I-3 : Schéma d'une boucle ouverte..... 4**

**Figure I-4: Symbole d'un sommateur..... 7**

**Figure I-5: Précision de Système..... 8**

**Figure I-6 :Précision de deux Systèmes apériodiques. ..... 8**

**Figure I-7: Rapidité de deux systèmes oscillatoires ..... 9**

**Figure I-8 :Stabilité du signal (température)..... 10**

**Figure I-9 : On/Off Controller (Changer la température)..... 13**

**Figure I-10: Schéma expliqué un réseau de neurones..... 14**

**Figure I-11 :Types d'ajustements d'un régulateur classique par réseaux de neurones..... 16**

**Figure II-1 : Asservissement par un régulateur PID ..... 19**

**Figure II-2 : Schéma bloc du système de 1er ordre avec des valeurs différentes de $kp$..... 22**

**Figure II-3 : La réponse du système avec $kp$..... 22**

**Figure II-4 : Modélisation sous Matlab de la réponse à un échelon dans un asservissement en position..... 23**

**Figure II-5 : Schéma bloc du système de 1er ordre avec des différentes valeurs de $ki$ ..... 24**

**Figure II-6 : La réponse du système avec $ki$..... 25**

**Figure II-7 : Réponse d'un système avec un dérivateur..... 26**

**Figure II-8 : Asservissement par régulateur PI. ..... 27**

Table des figures

**Figure II-9 : Modélisation sous Matlab de la réponse à un échelon dans un asservissement en vitesse..... 28**

**Figure II-10 : Asservissement par régulateur PD. ..... 29**

**Figure II-11 : Réponse indicielle du régulateur PD ..... 29**

**Figure II-12 : La structure parallèle ..... 30**

**Figure II-13 : La structure mixte..... 31**

**Figure II-14 : La structure série ..... 31**

**Figure II-15 : Schéma bloc de système de 3eme ordre commandé par les différentes structures de PID ..... 32**

**Figure II-16 : La réponse du système de 3eme ordre commandé par PID..... 32**

**Figure II-17 : Réponse d'un système avec tous les autres régulateurs ..... 33**

**Figure III-1 : La carte Arduino uno ..... 38**

**Figure III-2 : Microcontrôleur ATMega328P..... 40**

**Figure III-3 : Constitution de la carte Arduino UNO ..... 42**

**Figure III-4 :Brochage de la carte ArduinoUno..... 43**

**Figure III-5 : les entrées/sorites numérique de la carte Arduinouno..... 44**

**Figure III-6 : Les entrées analogiques de la carte Arduino uno..... 45**

**Figure III-7 : Schéma détaille du carte arduino ..... 47**

**Figure III-8 :structure d'un programme arduino ..... 50**

**Figure III-9 : schéma général du system..... 55**

**Figure III-10 : schéma détaillé du system..... 56**

**Figure IV-1 : Courbe de changement températureet l'erreur..... 61**

Table des figures

## Introduction générale

Les méthodes classiques de l'automatique ont été largement appliquées dans de nombreux problèmes de régulation industrielle. Cependant, la plupart des systèmes physiques présentent des non-linéarités et leurs paramètres sont souvent mal connus et/ou variables dans le temps. Pour la commande de telles classes de systèmes, les méthodes conventionnelles de l'automatique ont montré leurs limites en termes de stabilisation et performances. Suite aux développements des calculateurs numériques, les automaticiens commencent à s'intéresser aux nouvelles approches de commande telles que la commande adaptative, la commande prédictive, la commande robuste, ainsi que les techniques basées sur l'intelligence artificielle. [1]

Ce mémoire comporte quatre chapitres :

Le premier chapitre est devisé en quatre parties :

La première partie présente la notion de système

La deuxième partie présente une notion de l'automatique

La troisième partie présente une notion de régulation, asservissement et les types de régulation industrielle.

La dernière partie présente une notion Contrôles automatiques (PID Controller, On/Off Controller, Réseau de neurones)

Le deuxième chapitre est consacré à commande par le correcteur PID ainsi que les propriétés des actions proportionnelles, intégrateur et dérivateur et les différentes structures de PID et une méthode de calcul pour déterminer les paramètres de régulateur (P, I et D).

Le troisième chapitre est la conception (nous avons présenté notre travaille).

Dans le dernier chapitre est le résultat de travaille.

# Chapitre I

# Généralités sur les systèmes asservis

# Chapitre I. Généralités sur les systèmes asservis

## I.1. Introduction

L'automatique est une discipline scientifique qui vise à attribuer des propriétés souhaitées, à un dispositif donné, sans l'intervention de l'être humain. Pour ce fait, cette discipline passe par trois phases : une phase de modélisation (consiste à attribuer un modèle au comportement de ce dispositif), une phase d'analyse (permet de mieux comprendre ce comportement) et enfin une phase de commande (agir sur ce dispositif afin d'améliorer ses propriétés.

L'objet d'application de l'automatique est appelé système. Ce dernier doit être un objet qui bouge, qui se transforme ou qui fonctionne.

## I.2. Notion de système asservi

Un système peut être défini comme un ensemble d'éléments exerçant collectivement une fonction déterminée. Il est caractérisé par ses grandeurs d'entrée et de sortie.

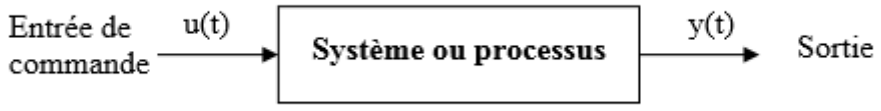


**Figure I-1 : Schéma bloc élémentaire**

Un système est dit asservi lorsqu'une grandeur de sortie suit la grandeur d'entrée (consigne) quelque soient les effets perturbateurs extérieurs.

- Lorsque la consigne d'un système asservi est indépendante du temps, on parle de régulation.
- Lorsque la consigne d'un système asservi dépend du temps, on parle d'asservissement (poursuite).

Le système asservi le plus simple correspond au schéma de la figure 1.2.

On distingue le processus, un actionneur qui est l'organe fournissant la puissance à partir du signal de commande élaboré par le correcteur, et un comparateur qui compare la valeur de la variable de sortie du processus mesurée par un capteur à la valeur de la consigne.

Cette structure fait intervenir deux chaînes, une chaîne d'action et une chaîne de retour. Ce type de système est appelé aussi système bouclé ou système en boucle fermée (BF). Dans le cas contraire, il a aucune rétroaction de la sortie du système sur son entrée, On parle de système non bouclé ou système en boucle ouverte (BO). La boucle fermée est capable de :

- Stabiliser un système instable en BO ;
- Compenser les perturbations externes ;
- Compenser les incertitudes internes au processus lui-même. La figure suivante présente le schéma fonctionnel d'un système asservi.

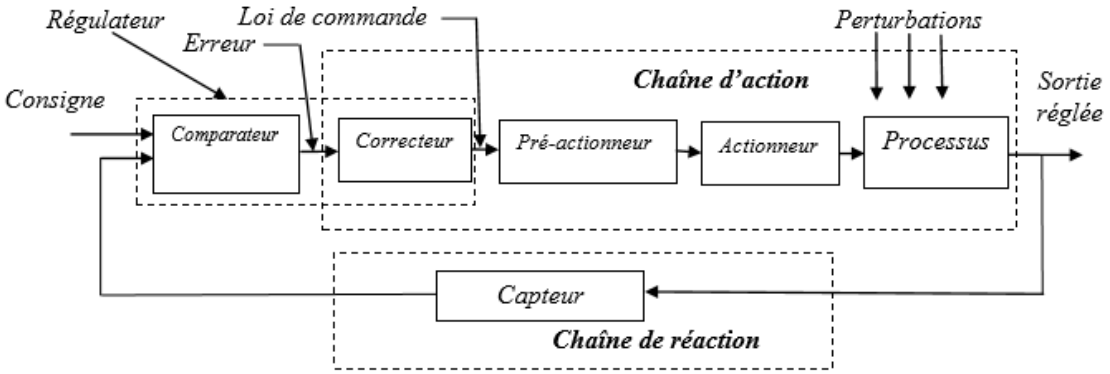


**Figure I-2: Schéma fonctionnel d'un système asservi**

## I.3. Chaîne de régulation automatique

### I.3.1. Système en boucle ouverte

Une cascade de sous-système constitue une boucle ouverte et une boucle fermée comporte souvent : - un actionneur - un processus à contrôler - un capteur

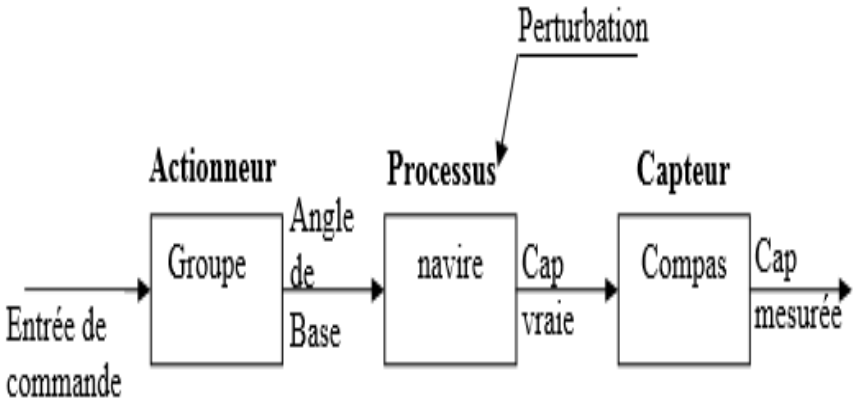


**Figure I-3 : Schéma d'une boucle ouverte**

On remarque pour le système en boucle ouverte que le signal de commande (entrée) est indépendant du signal de sortie.

### I.3.2. Système en boucle fermée

Un objectif majeur de l'automatique est la conception des lois de commande destinées à élaborer le signal de commande u(t) et ceci pour maîtriser un certain nombre de sorties de grandeurs physiques :

- Le courant ou la tension de sortie d'une source.
- La vitesse de rotation d'un moteur.
- La température d'un local.
- Le pilotage d'un navire.
- Ces lois sont mises en œuvre par des systèmes concrets analogiques numériques représentés par des schémas blocs.
- Un bloc de commande a pour sortie le signal de commande u(t) il a pour entrer d'une part le signal de consigne (référence) d'autre part le signal de sortie mesuré y(t).
- Le système global constitué du processus à contrôler et de système de commande constitue un système en boucle fermée.

### I.3.2.1. Principaux éléments d'une chaîne d'asservissement

Partie commande ou régulateur : le régulateur e compose d'un comparateur qui détermine l'écart entre la consigne et la sortie mesurée et d'un correcteur élabore à partir d'un signal d'erreur ε(t) l'ordre de commande de u(t) : C'est l'organe intelligent du système.

Actionneur : c'est l'organe d'action qui apporte l'énergie au système pour produire l'effet souhaité

Capteur : c'est l'organe qui prélève sur le système la grandeur asservie et la transforme en un signal compréhensible par le régulateur.

### I.3.2.2. Informations

**Entrée consigne :** La consigne et l'entrée de référence, c'est la grandeur régulant du système.

**Sortie régulée (asservie) :** la sortie régulée représente le phénomène que doit réguler. C'est la grandeur physique pour laquelle la sortie a été conçue.

**Perturbation :** on appelle perturbation tout phénomène physique intervenant sur le système qui modifie l'état de la sortie un système régulé doit pouvoir maintenir la sortie à son niveau indépendamment de la perturbation.

**Ecart (erreur) :** c'est la différence entre la consigne et la sortie. Cette mesure ne peut être réalisée que sur les grandeurs comparables. On la réalisera donc en général entre la consigne et la mesure de sortie

**Régulation et asservissement :**

- **Régulation :** On appelle régulation un système asservi(en BF) qui doit maintenir constante la sortie (conformément à la consigne et indépendamment des perturbations (ex : climatiseur, régulation de température…)
- **Asservissement :** On appelle asservissement un système asservi dont la sortie dépend (doit suivre) le plus fidèlement la consigne (consigne variable) (position : asservissement de position). [2]

### I.3.2.3. Buts et motivations d'un système asservi Un système automatique

Est un système capable d'effectuer plusieurs opérations sans intervention de l'homme, et qui ne peuvent lui être confié pour les raisons suivantes :

- Précision
- Caractère pénible des tâches à effectuer dans certains environnements.
- Complexité.
- Répétitivité

L'automatisation est également souvent une réponse de besoin d'amélioration de la productivité d'un produit.

## I.4. Modélisation

Un système qui est souvent représenté par son schéma physique peut le plus souvent être composé en des parties plus simples, ayant un nombre réduit de signaux d'entrée et de sortie. On fait appel alors aux connaissances physiques, électromécaniques, chimiques ou bien d'autres branches scientifiques pour écrire les équations qui régissent entre les entrées (causes) et les sorties (effets) : c'est le schéma fonctionnel. Après avoir décrit chaque élément, composant, partie ou sous système on aboutit à un système d'équations algébriques ou différentielles : Modèle mathématique.

### I.4.1. Schéma physique

Le schéma physique est un des représentations qui nous permettent d'analyser le système. Ce type de schéma utilise la normalisation de chaque technologie.

- **Schéma électrique** : circuit RLC.
- **Schéma mécanique** : masse, ressort, amortisseur…

### I.4.2. Schéma fonctionnel

Dans le schéma fonctionnel on représente les fonctions par des blocs et les grandeurs physiques par des flèches qui les relient. Lorsque la grandeur physique est obtenue par une sommation, on la représente par le symbole (+) et par un cercle.

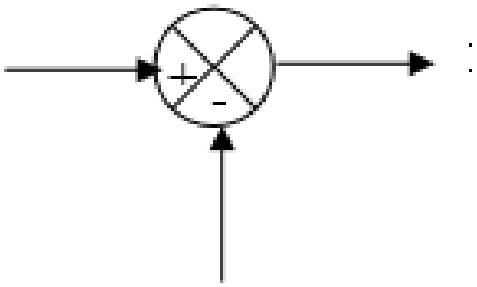


**Figure I-4: Symbole d'un sommateur**

### I.4.3. Modèle mathématique :

Pour réaliser une commande automatique il est nécessaire d'établir les relations existantes entre les entrées et les sorties. L'ensemble des relations s'appelle modèle mathématique d'un système.

E(t)=Ur(t)+Ul(t)+Vs(t)

Résistance : Ur(t) =Ri(t)

Bobine : Ul(t)=L*di/dt

Condensateur :Vs (t)= (1/c)*∫i(t)*dt

## I.5. Caractéristiques dynamiques d'un système asservi

Les caractéristiques dynamiques d'un système asservi permettent de quantifier les performances d'un système asservi. Elles sont appréciées à partir de la réponse des systèmes des entrées types.

### I.5.1. Performances d'un système asservi

#### A. Précision

La précision quantifie l'erreur lorsque l'équilibre est atteint avec e(t) et s(t) sont de même nature, autrement l'erreur se trouve à la sortie du comparateur.

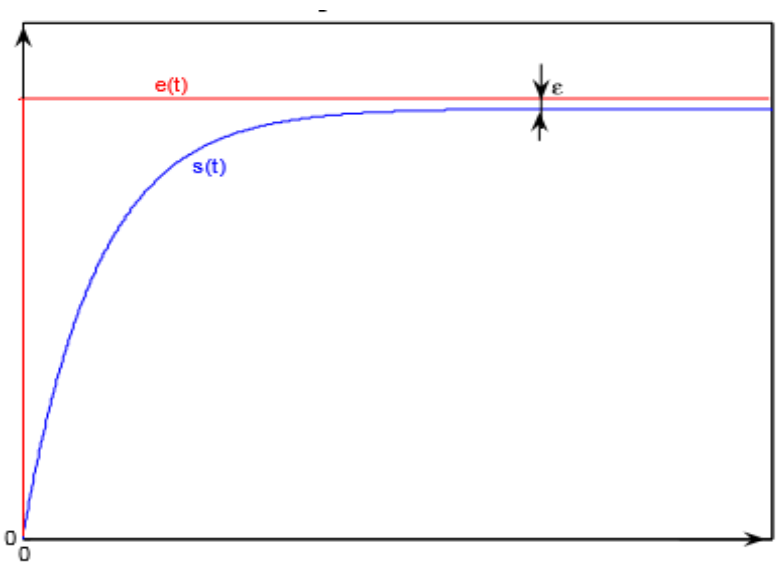


**Figure I-5: Précision de Système**

Un système est précis si la sortie suit l'entrée à toutes circonstances.

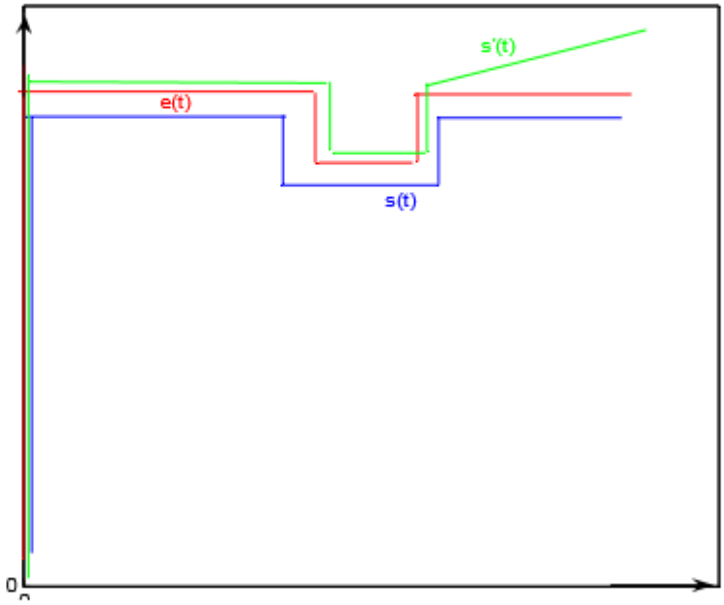


**Figure I-6 :Précision de deux Systèmes apériodiques.**

- Le signal s(t) : précis
- Le signal s'(t) : s n'est pas précis

## B. Rapidité

La rapidité quantifie le temps pour atteindre l'équilibre on l'appelle le temps de réponse. Le temps mis par la réponse du système pour atteindre à moins de 5%, la valeur finale est retenue comme critère de rapidité.

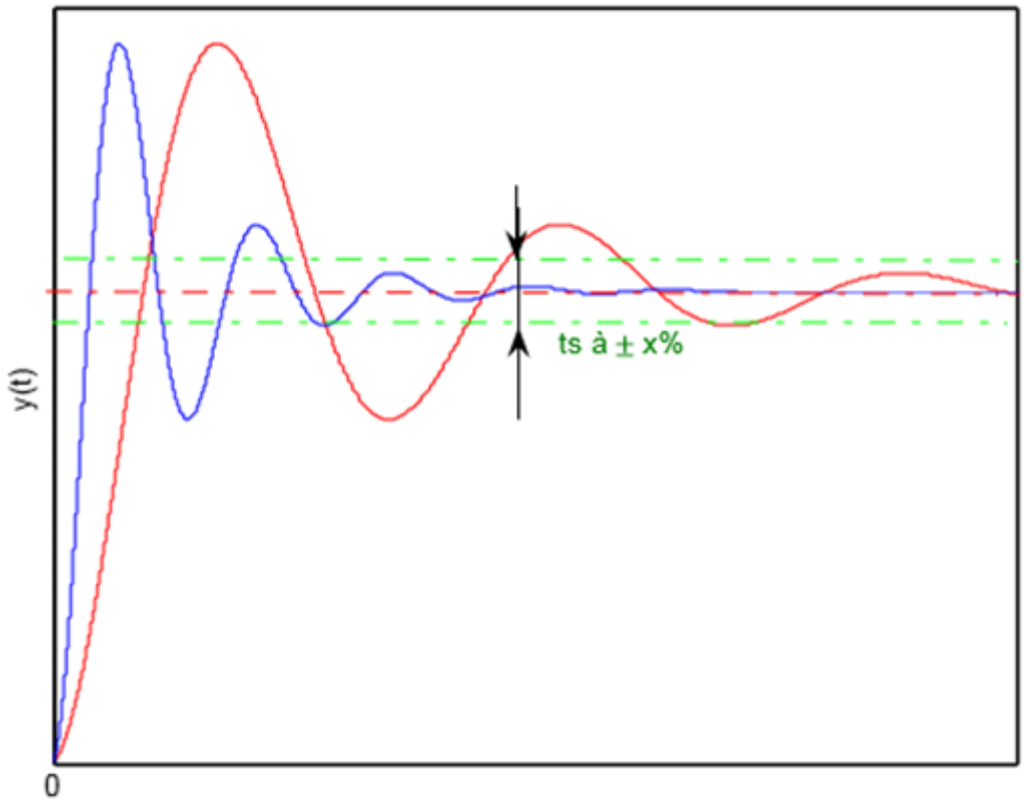


**Figure I-7: Rapidité de deux systèmes oscillatoires**

Le signal s(t) est plus rapide que le signal s'(t).

On dit alors qu'un système a une rapidité satisfaisante s'il se stabilise à son niveau constant en un temps jugé satisfaisant.

## C. Stabilité

- La stabilité d'un système est la capacité de converger vers une valeur constante si l'entrée est constante.

- Ce système tend à revenir à son état d'équilibre permettant quand on lui applique une perturbation de courte durée.

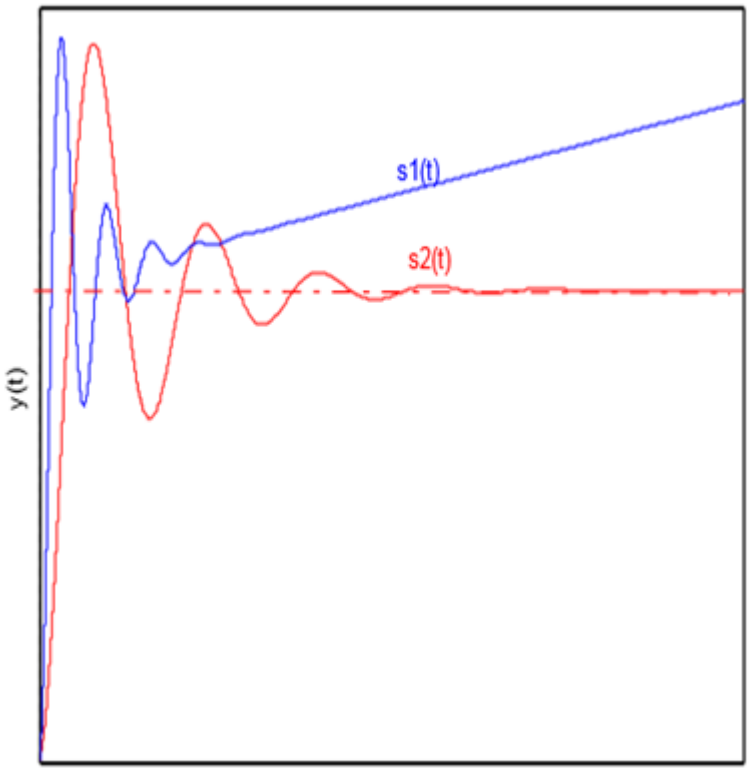


**Figure I-8 :Stabilité du signal (température)**

- Le signal s1(t) : stable
- Le signal s2(t) : instable

## I.6. Contrôles automatiques

### I.6.1. PID Controller

Un contrôleur dérivé proportionnel-intégral (contrôleur PID) est un retour de boucle de contrôle Mécanisme (contrôleur) largement utilisé dans les systèmes de contrôle industriels. Un contrôleur PID calcule une valeur d'erreur correspondant à la différence entre une variable de processus mesurée et un point de consigne souhaité.

Le contrôleur tente de minimiser l'erreur en ajustant le processus en utilisant un variable manipulée L'algorithme du contrôleur PID implique trois paramètres constants distincts et est en conséquence parfois appelé contrôle à trois termes : les valeurs proportionnelle, intégrale et

dérivée notés P, I et D. En termes simples, ces valeurs peuvent être interprétées en termes de temps :

Dépendeur l'erreur présente, I sur l'accumulation d'erreurs passées et D est une prédiction d'erreurs futures, en fonction du taux de variation actuel. La somme pondérée de ces trois actions est utilisé pour ajuster le processus via un élément de contrôle tel que la position d'une vanne de contrôle, un registre ou la puissance fournie à un élément chauffant.

### I.6.1.1. Proportionnel

Le premier composant de l'algorithme PID est le plus simple à comprendre et le plus crucial pour les performances du contrôleur. Le P signifie proportionnelle. Cela signifie que la variable de contrôle doit être ajustée proportionnellement à la quantité d'erreur dans le système. Un algorithme PID utilisant uniquement une composante P pourrait être exprimé par : sortie = erreur * Kp

- **Variable de contrôle :** La variable de contrôle est la sortie du contrôleur que nous devons ajuster.
- **Variable de processus :** La variable de processus est la valeur mesurée dans le système que vous tentez de contrôler. La variable de processus sert de retour d'information au contrôleur pour lui permettre de décider comment l'ajuster.
- **Erreur :** L'erreur correspond à la différence entre la variable de processus et le point de consigne.

La composante P de l'algorithme fonctionne en ajustant la sortie proportionnellement à l'erreur

### I.6.1.2. Intégral

Est un terme mathématique qui signifie accumuler quelque chose avec le temps *. Dans le cas des PID, le composant I accumule l'erreur survenue dans le temps. Cette erreur accumulée est ensuite multipliée par Ki, le coefficient intégral, et ajoutée à la sortie.

accumulation_de_erreur + = erreur * delta_time

Sortie = (erreur * Kp) + (accumulation_de_erreur * Ki)

La composante intégrale de l'algorithme de contrôle peut supprimer toute erreur d'état stable dans le système car elle accumule cette erreur dans le temps et la compense, plutôt que de simplement regarder un instantané de l'erreur à un moment donné.

### I.6.1.3. Dérivé

Le D signifie dérivé et est probablement le plus compliqué. Le composant dérivé est utilisé moins fréquemment dans les contrôleurs mais reste important dans certaines applications.

Le composant dérivé est responsable de la compensation des changements soudains de l'erreur.

Par exemple Si nous jetions de la glace dans notre marmite d'eau, la température chutait soudainement, l'erreur augmentait soudainement et la composante dérivée de l'algorithme entrait en jeu et augmentait le rendement du brûleur.

### I.6.2. On/Off Controller

La régulation peut être effectuée manuellement mais qui nécessite un personnel important et n'est pas particulièrement précis, donc,

Automatique le contrôle est utilisé aujourd'hui. Pour prévenir les blessures à Personnel et dommages à la réfrigération matérielle, les plantes doivent être équipées

Contrôles de sécurité. Danfoss régulant

L'équipement couvre ces exigences. Les offres suivantes avec ces réfrigérations

Contrôles qui règlent conformément à le principe marche / arrêt.

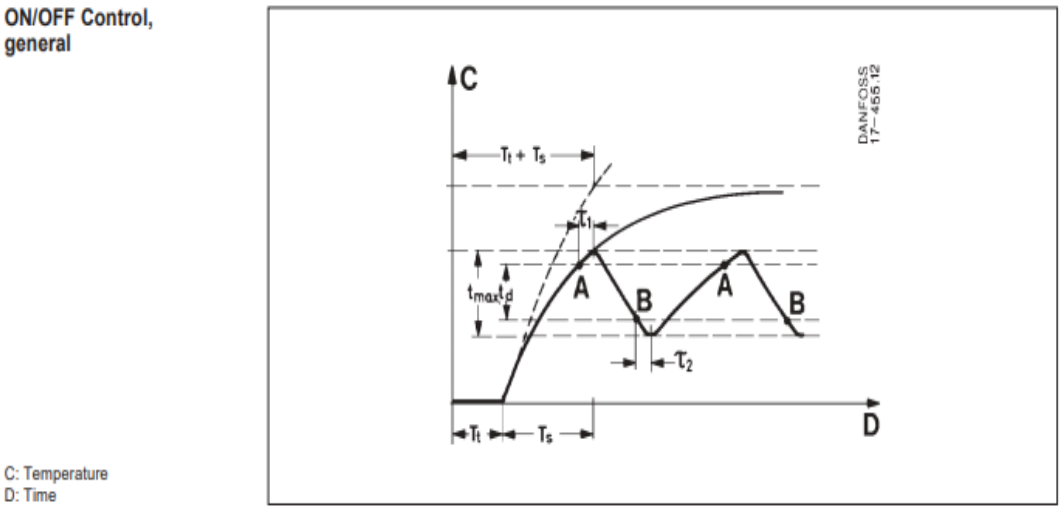


**Figure I-9 : On/Off Controller (Changer la température)**

Le but du contrôle marche / arrêt est de garder un donnée physique donnée, par ex. l'ambiant température, dans certaines limites ou à changer il selon un programme prédéterminé.

Un système de contrôle sert à mesurer la valeur de la variable contrôlée, comparez-le avec le la valeur souhaitée et réglez l'unité de commande en lequel une déviation possible est réduite.

Thermostats et contrôles de pression pour marche / arrêt le contrôle sont des régulateurs à deux positions où la variable manipulée ne peut conduire à deux

Conditions: cut-in ou cut-out.

### I.6.3. Réseau de neurones

#### I.6.3.1. Définition d'un réseau de neurones

Ensemble de neurones formels interconnectés permettant la résolution de problèmes complexes tels que la reconnaissance des formes ou le traitement du langage naturel, grâce à l'ajustement des coefficients de pondération dans une phase d'apprentissage.

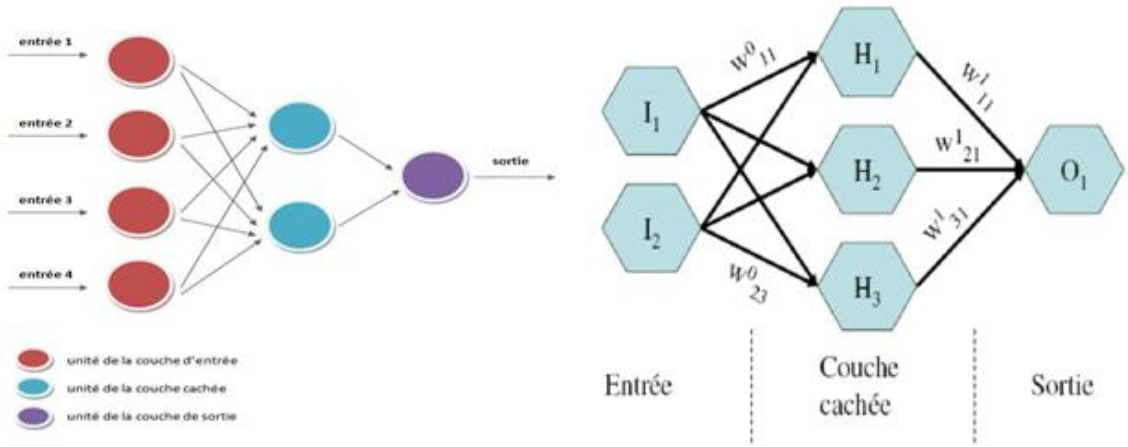


**Figure I-10: Schéma expliqué un réseau de neurones**

### I.6.3.2. Fonctionnement du réseau neuronal

Un réseau neuronal s'inspire du fonctionnement des neurones biologiques et prend corps dans un ordinateur sous forme d'un algorithme. Le réseau neuronal peut se modifier lui-même en fonction des résultats de ses actions, ce qui permet l'apprentissage et la résolution de problèmes sans algorithme, donc sans programmation classique.

### I.6.3.3. La relation entre réseau de neurones et PID

Autres types d'architectures de commande neuronale plus simples sont exposés. Elles sont basées sur l'apprentissage d'un contrôleur conventionnel déjà inséré dans la boucle de commande. La Figure.1.11regroupe quatre architectures de commande. Le régulateur conventionnel de type proportionnel-intégral-dérivé (PID) est pris comme exemple dans les différentes illustrations.

Le schéma de commande par identification directe d'un régulateur est illustré par la Figure.1.11.a. Dans ce schéma, un RNI est illustré pour faire une identification hors ligne du régulateur. Une fois cette identification est accomplie, le RNI remplacera le régulateur conventionnel dans la boucle de commande et fonctionnera en tant que RNC. Cette méthode trouve son intérêt lorsqu'on veut s'affranchir des contraintes liées à l'implémentation des régulateurs conventionnels.

La commande par apprentissage en parallèle avec un régulateur représentée par le schéma de la Figure.1.11.b est constituée d'un RNC qui fonctionne en ligne et en parallèle avec le régulateur PID. Le RNC réalise un apprentissage en ligne grâce à l'erreur calculée à partir de la consigne R(k) et la sortie du processus yd(k). La sortie du RNC, U2(k), est additionnée avec la sortie U1(k) du régulateur conventionnel afin de la corriger. L'intérêt de ce schéma est que le RNC opère en ligne et permet de corriger les insuffisances du régulateur PID notamment lors du changement des paramètres du processus.

Un autre schéma consiste en la commande par apprentissage d'un régulateur est illustré par Figure.1.11.c. Dans cette configuration, le RNC est corrigé par la sortie U2(k) du régulateur conventionnel dans le but de minimiser cette sortie et donc d'éliminer son effet dans la boucle de commande. Après apprentissage, donc U2(k) tend vers une très faible valeur, le RNC sera utilisé pour la commande du processus en boucle ouverte.

La dernière stratégie présentée est la commande par auto-ajustement des paramètres d'un régulateur PID illustré sur par la Figure.1.11.d. Dans ce schéma, les paramètres du régulateur PID (Kp, Ki et KD) sont déterminés en ligne par le RNC. Ensuite, ils sont injectés dans la structure du PID afin de procéder à la régulation du processus. Cette architecture offre la caractéristique adaptative à la structure du régulateur conventionnel. Ceci permettra aux paramètres du régulateur

PID de suivre en temps réel les changements des paramètres du processus.

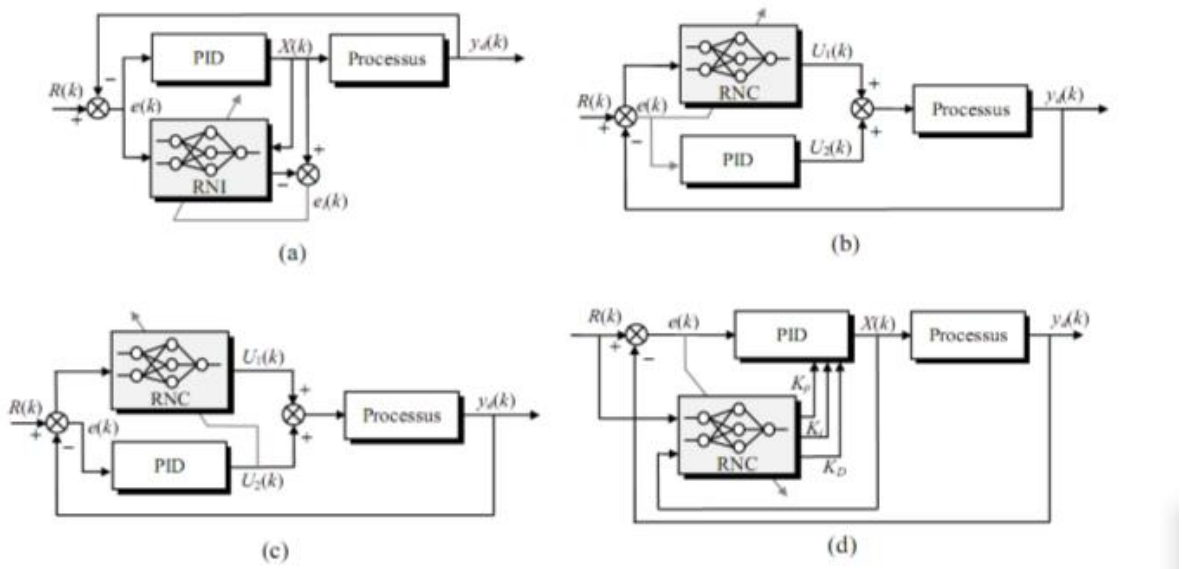


**Figure I-11 :**Types d'ajustements d'un régulateur classique par réseaux de neurones

### I.6.3.4. Application des réseaux des neurones dans l'industrie

Vu ses avantages les réseaux de neurones sont plus en plus utilisés dans l'industrie. Dans cette section nous citons quelques applications industrielles utilisant les réseaux neurones. Pour plus de détails sur les méthodes d'application, l'étudiant est invité à consulter les références de notre cours de chaque application tel que :

- la commande des systèmes électriques ;
- le traitement des eaux ;
- l'identification des systèmes ;
- la reconnaissance des formes.

### A. Avantage

Les principales qualités des réseaux de neurones sont leur capacité d'adaptabilité et d'auto organisation et la possibilité de résoudre des problèmes non-linéaires avec une bonne approximation. Ils ont une bonne immunité aux bruits et se prêtent bien à une implantation parallèle. La rapidité d'exécution est une qualité importante et elle justifie souvent à elle seule le choix d'implanter un réseau de neurones. Ces qualités ont permis de réaliser avec succès, plusieurs applications : classification, filtrage, compression de données, contrôleur

### B. Inconvénients

La difficulté d'interpréter le comportement d'un réseau de neurones est un inconvénient pour la mise au point d'une application. Il est souvent impossible d'utiliser les résultats obtenus pour améliorer ce comportement. Il est également hasardeux de généraliser à partir d'expériences antérieures et de conclure ou de créer des règles sur le fonctionnement et le comportement des réseaux de neurones. Beaucoup d'heuristiques sont utilisées, mais elles se contredisent parfois et elles ne permettent pas toujours de trouver des valeurs optimales

## I.7. Conclusion

Dans ce chapitre nous avons vu la notion de système et leur performance, ensuite on vu la définition d'un système asservis et les types de régulateurs industriels, poursuite on définir la modélisation et la fonction de transfert des systèmes asservis. Dans le chapitre suivant, nous allons identifier les propriétés d'un régulateur plus utilisé dans l'industrie (PID)

# Chapitre II :

# PID

# Chapitre II. PID

## II.1. Introduction

Un contrôleur PID est un instrument utilisé dans les applications de contrôle industriel pour réguler la température, le débit, la pression, la vitesse et d'autres variables de processus. Les contrôleurs PID (dérivées proportionnelles intégrales) utilisent un mécanisme de rétroaction de boucle de contrôle pour contrôler les variables de processus et constituent le contrôleur le plus précis et le plus stable.

## II.2. Comment Travailler

Nous allons considérer le système de retour d'unité suivant:

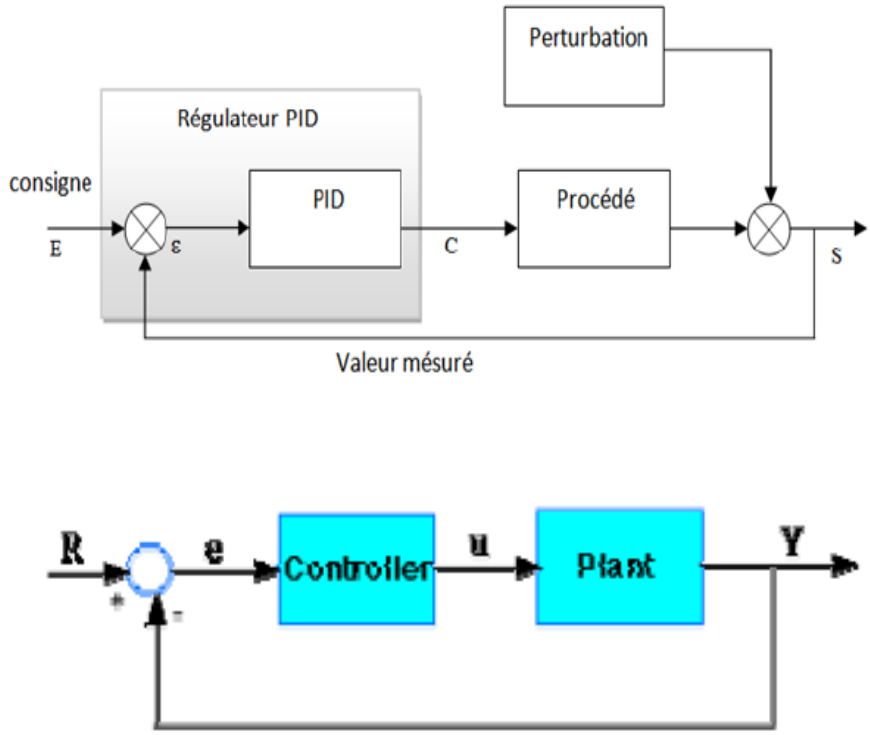


**Figure II-1 : Asservissement par un régulateur PID**

Plant : Un système pour contrôler

Contrôleur : fournit l'enthousiasme pour la plante ; Conçu pour contrôler l'ensemble

Comportement du système

**Le contrôleur à trois termes**

La fonction de transfert du contrôleur PID se présente comme suit :

- $Kp+ (Ki/S)+Kd*S=(Kp*S+Kd*S^2+Ki)/S$
- Kp = Proportional gain
- KI = Integral gain
- Kd = Dérivative gain

Voyons d'abord le fonctionnement du contrôleur PID dans un système en boucle fermée utilisant le schéma présenté ci-dessus. La variable (e) représente l'erreur de suivi, la différence entre la valeur d'entrée souhaitée (R) et la sortie réelle (Y). Ce signal d'erreur (e) sera envoyé au contrôleur PID, et le contrôleur calcule à la fois le dérivé et l'intégrale de ce signal d'erreur. Le signal (u) juste après le contrôleur est maintenant égal au proportionnel gain (Kp) fois la magnitude de l'erreur plus le gain intégral (Ki) fois l'intégrale de l'erreur plus le gain dérivé (Kd) fois la dérivée de l'erreur.

## II.3. Les Régulateurs

- Il-y-a deux type de régulateur : Les Régulateurs simples
- Les Régulateurs complexes

### II.3.1. Les Régulateurs simples

Utilisé une seule méthode comme :

- Régulateur à action proportionnel (P)
- Régulateur à action intégrale (I)
- Régulateur à action dérivée (D)

### II.3.1.1. Régulateur à action proportionnel (P)

Le régulateur à action proportionnelle, ou régulateur P, a une action simple, puisqu'il construit une commande $kp$ proportionnelle à l'erreur e(t). Ou l'erreur est virtuellement amplifiée d'un certain gain constant qu'il conviendra de déterminer en fonction du système.

- C (t) =$kp$.e(t)

Ce qui en Laplace donne :

- C (p) = $kp$.e(p)

L'idée étant d'augmenter l'effet de l'erreur sur le système afin que celui-ci réagisse plus rapidement aux changements de consignes. Plus la valeur de $kp$ est grande, plus la réponse l'est aussi. la stabilité du système s'en trouve détériorée et dans le cas d'un $kp$ démesuré le système peut même diverger.

Si l'on prend l'exemple d'un système de 1er ordre avec des différentes valeurs du gain $kp$.

**Remarque** : Nous pouvons utiliser la lettre S à la place de lettre P.

- P formes françaises.
- S formes américaines.

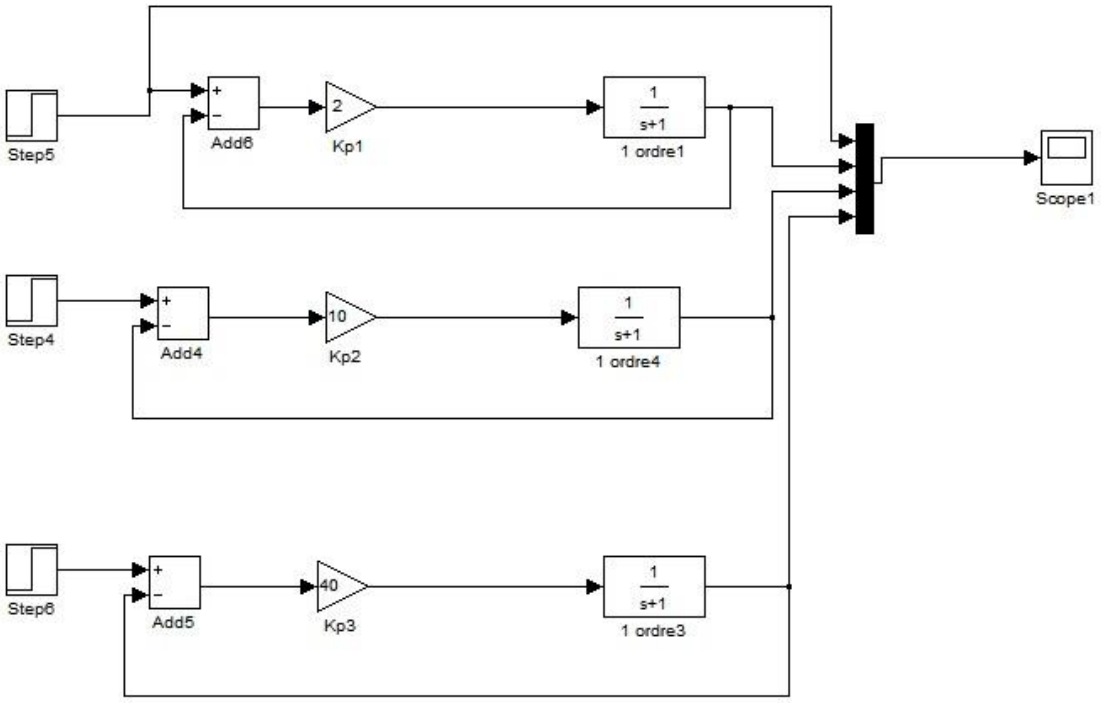


Figure II-2 : Schéma bloc du système de 1er ordre avec des valeurs différentes de ***kp***

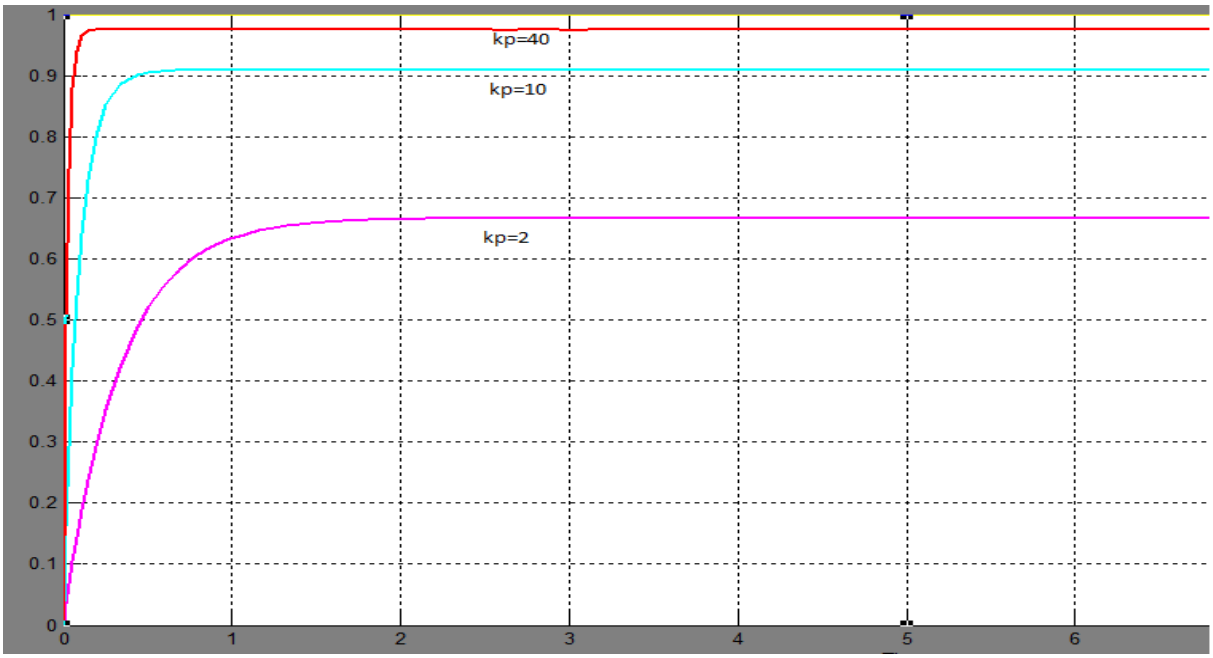


**Figure II-3 :** La réponse du système avec ***kp***

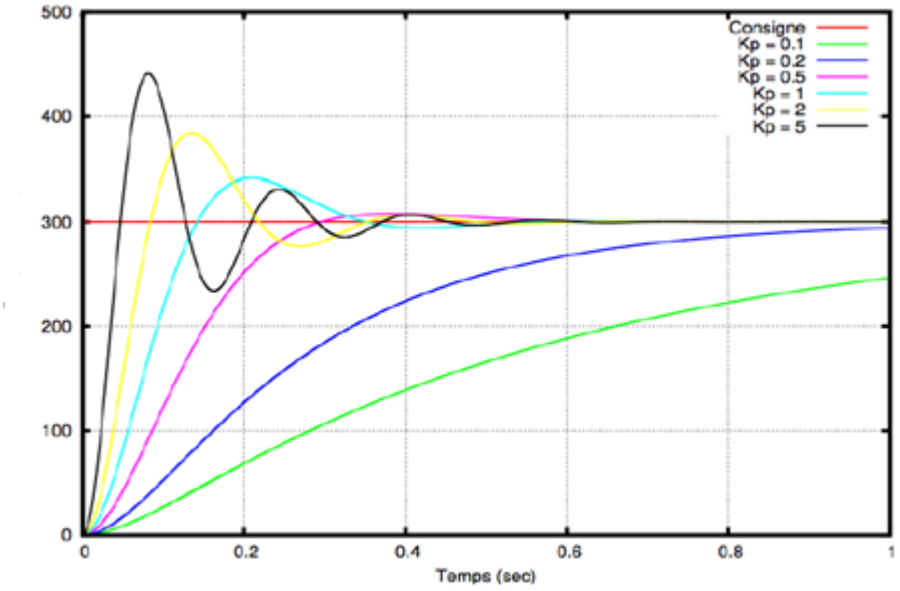


**Figure II-4 :** Modélisation sous Matlab de la réponse à un échelon dans un asservissement en position.

**Remarque :** on observe que si le gain $kp$ augmenté, l'erreur statique et le temps de réponse sont diminuée et le dépassement aussi augmenté.

### II.3.1.2. Régulateur à action intégrale (I)

Les exemples des asservissements vus précédemment ont montré qu'un système, même contre-réaction par un régulateur P, pouvait présenter une erreur permanente en régime permanent constant. Cette erreur intervenant alors que les signaux d'entrée (consigne ou perturbation) sont constants, on la désigne par erreur statique.

Pour remédier au problème du statisme la solution consiste à intégrer l'erreur, la loi de commande est de la forme :

- $C(t) = ki.\int(\tau)t0.d\tau = 1/Ti\ .\int e(\tau)t0.d\tau$ Ou $ki = 1/Ti$

Ce qui en Laplace donne :

- $C(t) = ki.\ (p)/p = 1/Ti.p.\ e(p)$

La constante de temps $Ti$ exprimée souvent en unité de temps est appelée la constante de temps d'intégration.

Si l'on prend l'exemple d'un système de 1er ordre avec des différentes valeurs du gain $ki$.

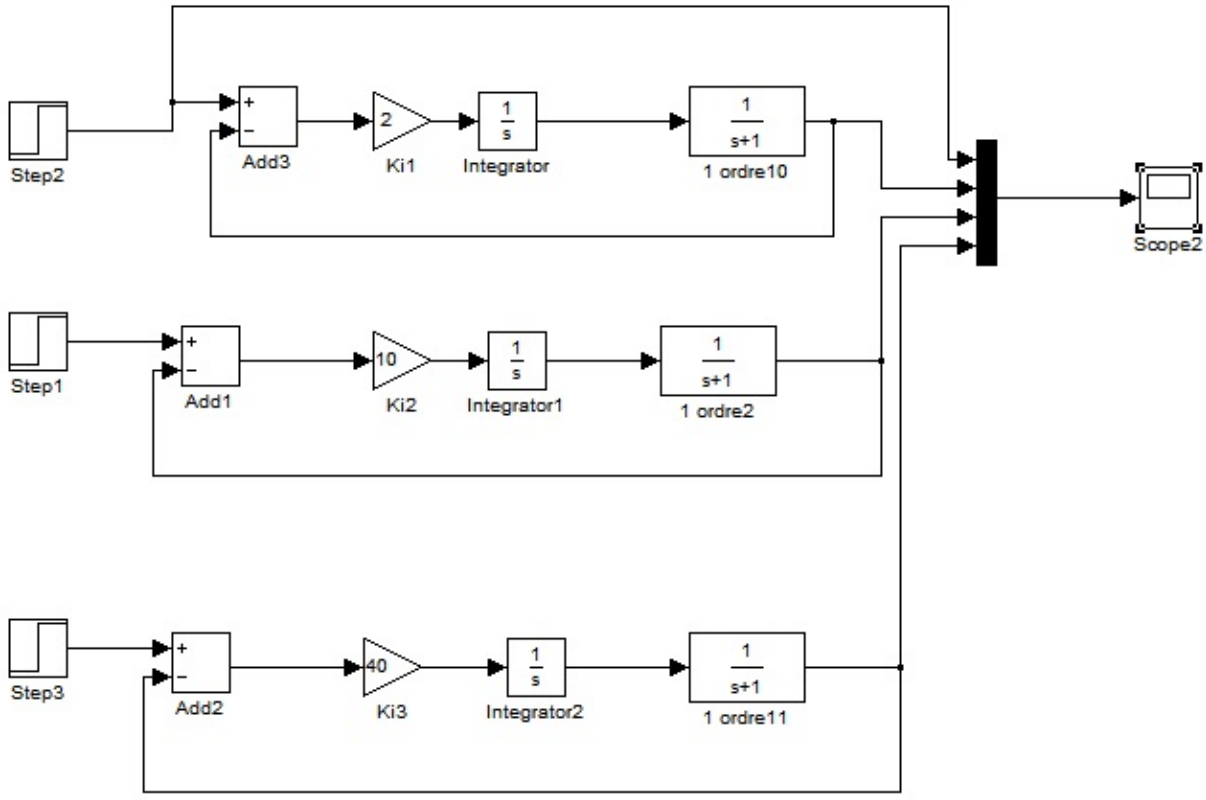


**Figure II-5 :** Schéma bloc du système de 1er ordre avec des différentes valeurs de ***ki***

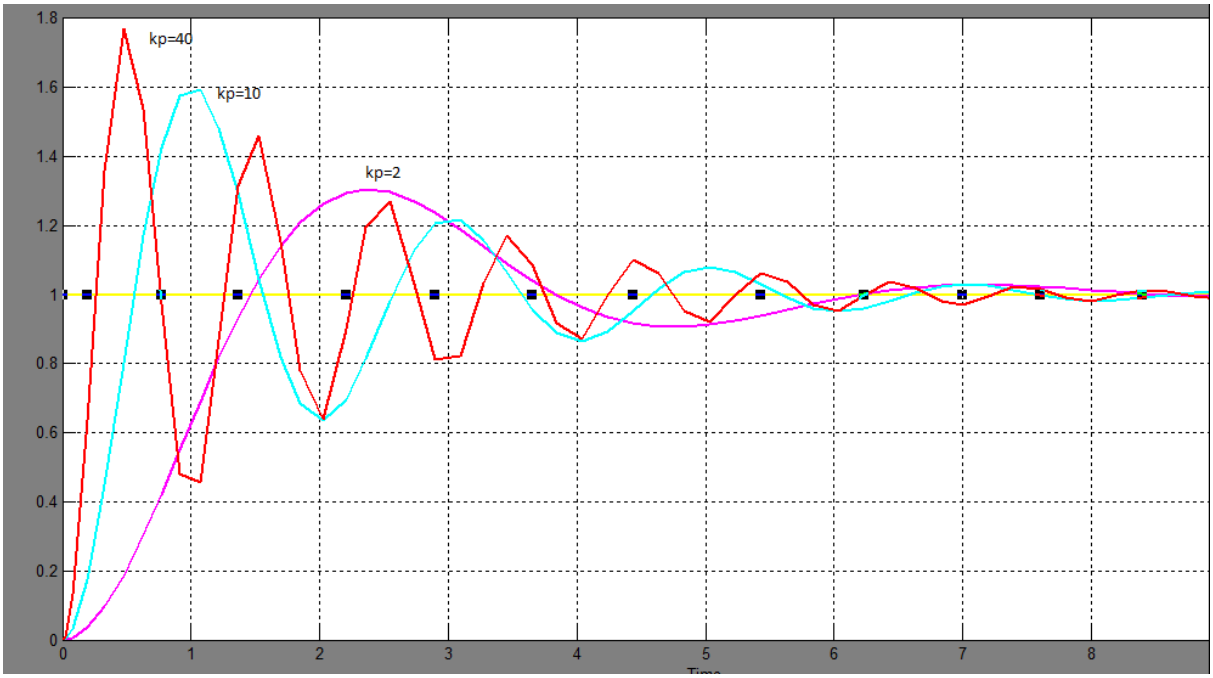


**Figure II-6 :** La réponse du système avec $\boldsymbol{ki}$

**Remarque** : on observe que si on augmente le gain $ki$ (diminue la constante de temps d'intégration $Ti$), l'erreur statique est nulle en régime permanent quelle que soit cette valeur, Et il fait un peu changement de dépassement, plus la stabilité se dégrade, et accélère la réponse.

Dans l'industrie, on utilisera l'action I chaque fois qu'on a besoin, pour des raisons technologiques, d'avoir une précision parfaite.

**Exemple** : la régulation de la pression ou de la température dans un réacteur nucléaire.

### II.3.1.3. Régulateur à action dérivée (D)

Pourquoi pouvons-nous avoir besoin d'un terme dérivé ?

Et bien, le contrôle P et I peut amener à un dépassement de la consigne, ce qui n'est pas toujours très souhaitable (exemple d'inversion de polarité dans le cas des moteurs électriques). Le terme dérivé permet de limiter cela. Lorsque le système s'approche de la consigne, ce terme freine le système en appliquant une action dans le sens opposé et permet ainsi une stabilisation plus rapide.

La loi de la commande est de la forme :

- $C(t) = kd.(de(t)/dt) = Td.\ (de(t)/dt)$

Soit d'après la transformée de Laplace :

- $C(p) = kd.\ P.\ e(p) = Td.P.\ e(p)$

Où

- $kd = Td$La constante de temps d'action dérivée notée $Td$exprimée en seconde.

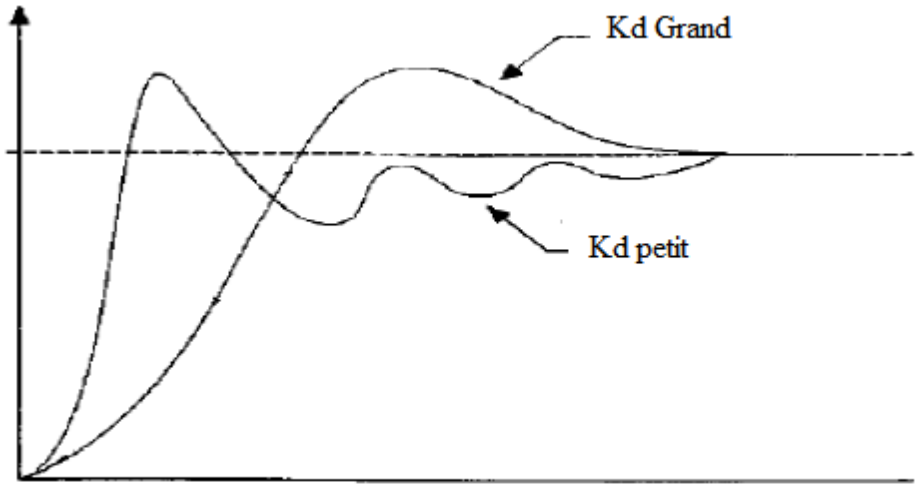


Figure II-7 : Réponse d'un système avec un dérivateur

**Remarque :** on observe que si nous augmentons le gain $kd$ le temps de réponse est diminué et le dépassement est réduit, et par un excès d'action dérivée qui peut conduire à l'instabilité du système bouclé.

Pour déstabiliser Si vous avez un système comme une boucle de température très lent, les utilisateurs ont tendance à mettre beaucoup de dériver là parce qu'ils n'aiment pas le dépassement," dit-il.

### II.3.2. Les Régulateurs complexes

Elles utilisé ou mois deux méthodes comme :

- Régulateur PI
- Régulateur PD
- Régulateur PID

#### II.3.2.1. Régulateur PI

Au contrôle proportionnel, nous pouvons ajouter l'intégration de l'erreur. Dans ce cas nous obtenons une régulation PI (proportionnelle et intégré).

La loi de commande est de la forme :

- $C(t) = kp.e(t) + k.\int(\tau)t0.d\tau$

Soit d'après la transformée de Laplace :

- $C(p) = kp.e(p) + ki.\ e(p)/p$

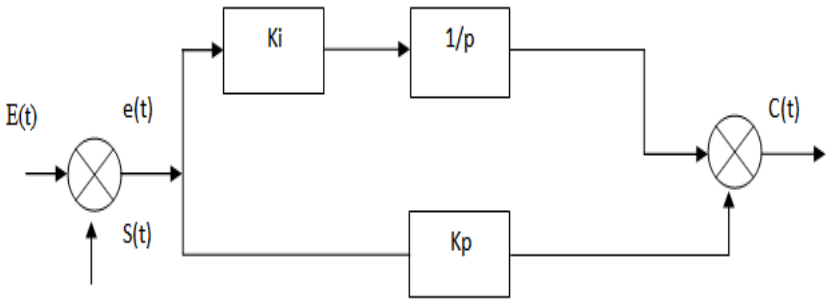


**Figure II-8 :** Asservissement par régulateur PI.

Pourquoi a-ton besoin de rajouter cette fonctionnalité à notre organe de contrôle ?

Et bien, lors d'un simple contrôle proportionnel, il subsiste une erreur statique. Lorsque le système s'approche de sa consigne, l'erreur n'est plus assez grande pour faire avancer le moteur. Le terme intégral permet ainsi de compenser l'erreur statique et fournit, par

Conséquent, un système plus stable en régime permanent. Plus $ki$ est élevé, plus l'erreur statique est corrigée.

Pour reprendre l'exemple de la voiture qui dérive, le terme intégral consiste à rajouter un petit Coup de contre braquage afin de se rétablir correctement.

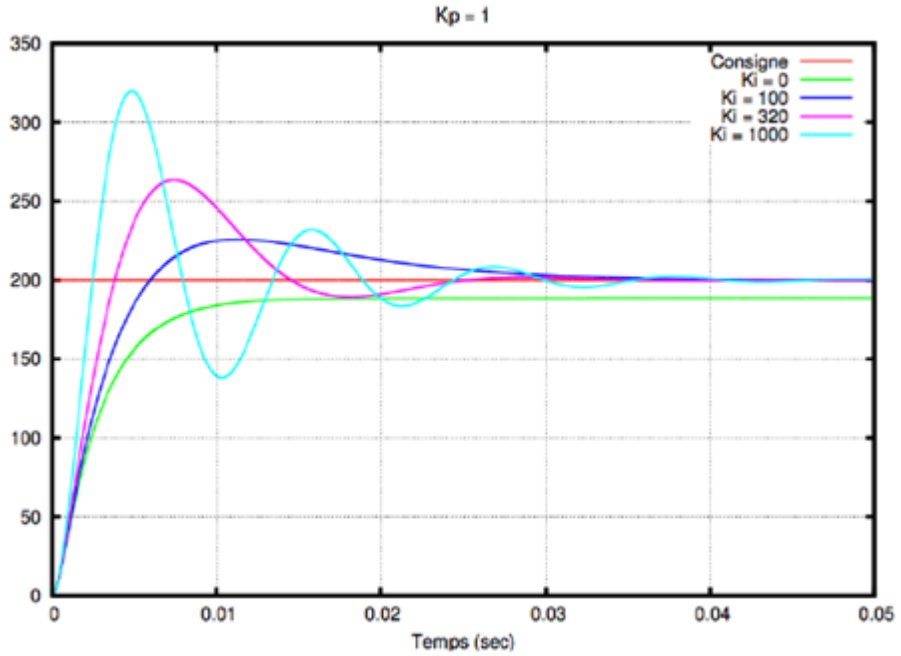


**Figure II-9 :** Modélisation sous Matlab de la réponse à un échelon dans un asservissement en vitesse

Le régulateur PI est le régulateur le plus utilisé en pratique où ses contributions à la précision mais aussi à la robustesse du système asservi sont particulièrement appréciées.

### II.3.2.2. Régulateur PD

On notera que l'action D ne permettant pas la transmission d'un signal constant, elle doit donc toujours s'accompagner au moins d'une action P en parallèle.

La loi de commande est de la forme :

- $C(t) = kp.e(t) + k.(t)dt$

Soit d'après la transformée de Laplace :

- $C(p) = kp.e(p) + kd.\ P.\ e(p)$

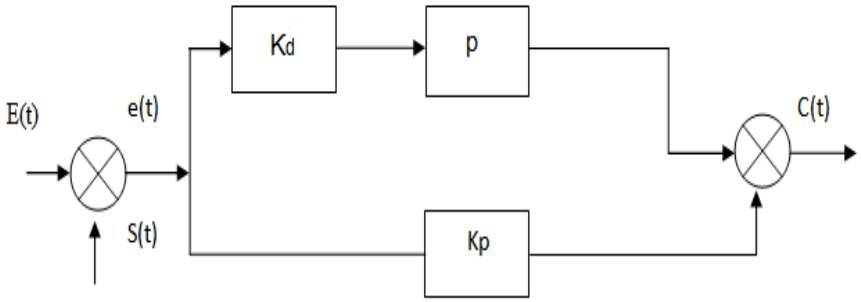


**Figure II-10 :** Asservissement par régulateur PD.

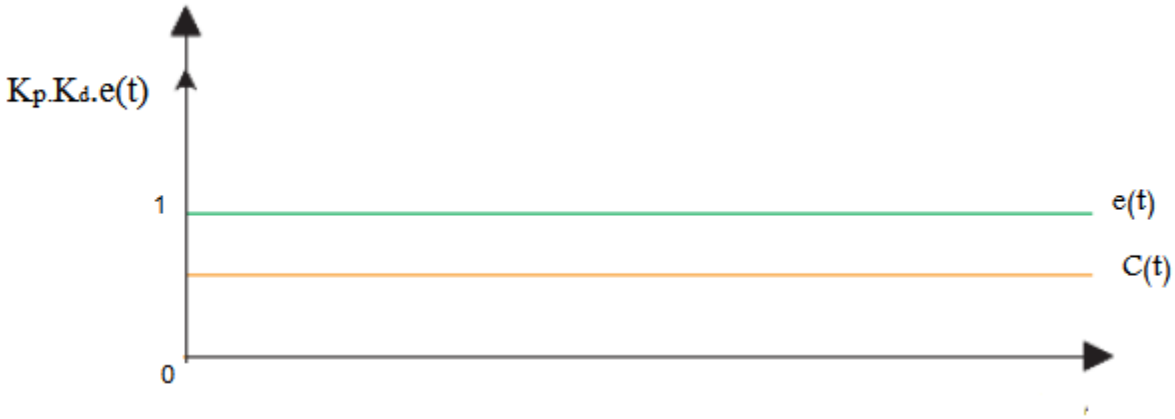


**Figure II-11 :** Réponse indicielle du régulateur PD

Régulateur PD effet stabilisant et amélioration de la rapidité mais il a des Inconvénients de sensibilité aux bruits et précision statique.

### II.3.2.3. Régulateur Proportionnel Intégrateur Dérivé PID

PID est un régulateur qui support des trois actions P, I et D. il caractérise par réunir les effets positifs des trois correcteurs de base (Proportionnel, Intégrateur et Dérivateur). Avantage de terme I, il permet l'annulation d'une erreur statique tout en autorisant avantage

de l'action D des performances de rapidité à celles d'un régulateur PI.La loi de commande est de la forme :

- $C(t) = kp.e(t) + k.\int(\tau)t0.d\tau + kd.\ de(t)/dt$

Soit d'après la transformée de Laplace :

- $C(p) = kp.e(p) + k.(p)p + kd.$ P. e(p)

## A. Les différentes structures du PID

### La structure parallèle

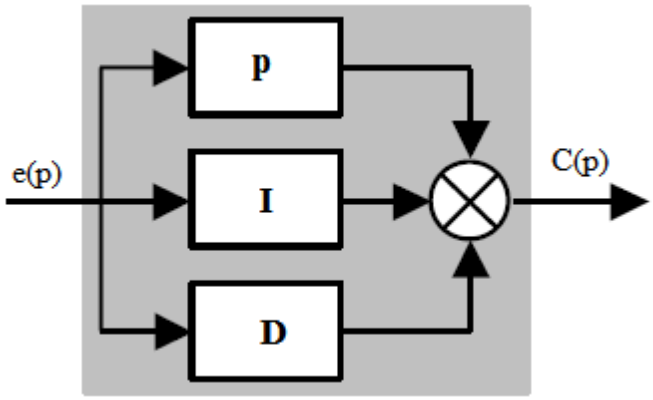


**Figure II-12 :** La structure parallèle

La loi de commande est de la forme :

- $C(t) = kp.e(t) + k.\int(\tau)t0.d\tau + kd.\ de(t)/dt$

Soit d'après la transformée de Laplace :

- $C(p) = kp.e(p) + k.(p)p + kd.\ P.\ e(p)$

C’est la structure le plus utile dans l’industrie.

**La structure mixte**

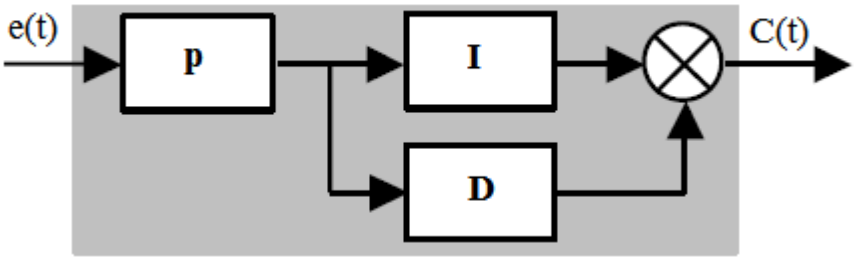


**Figure II-13 :** La structure mixte

Soit d’après la transformée de Laplace, La loi de commande est de la forme :

- $C(p) = kp.e(p).[ki.\ 1p + kd.\ P]$

**La structure série**

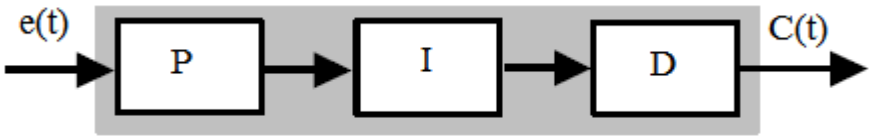


**Figure II-14 :** La structure série

Soit d’après la transformée de Laplace, La loi de commande est de la forme :

- $C(p) = e(p).kp.\ [k.1].[kd.\ P]$

## B. Schéma bloc

Le schéma bloc du system avec des différentes structures de PID, On prend un exemple du système de 3éme ordre de la fonction de transfert suivant :

$(p)=6p3+6p2+11p+6$

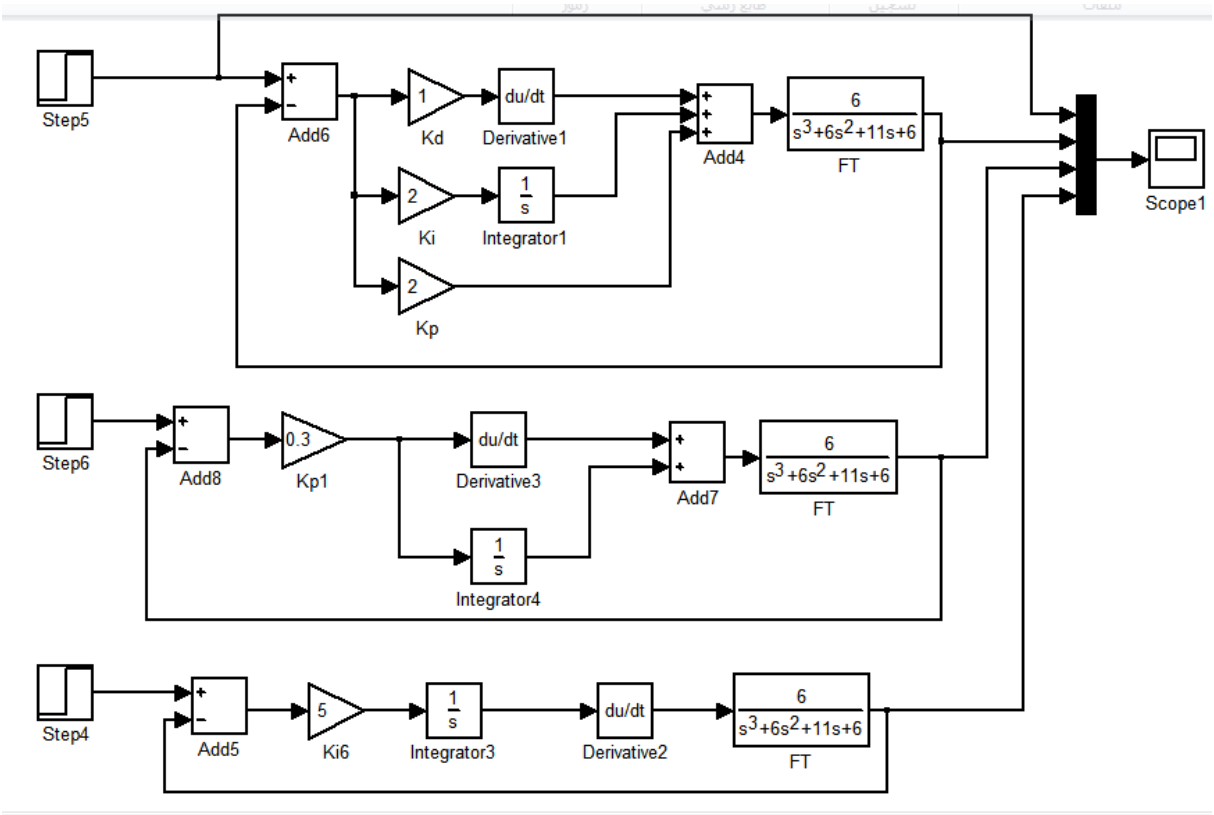


**Figure II-15 :** Schéma bloc de système de 3eme ordre commandé par les différentes structures de PID

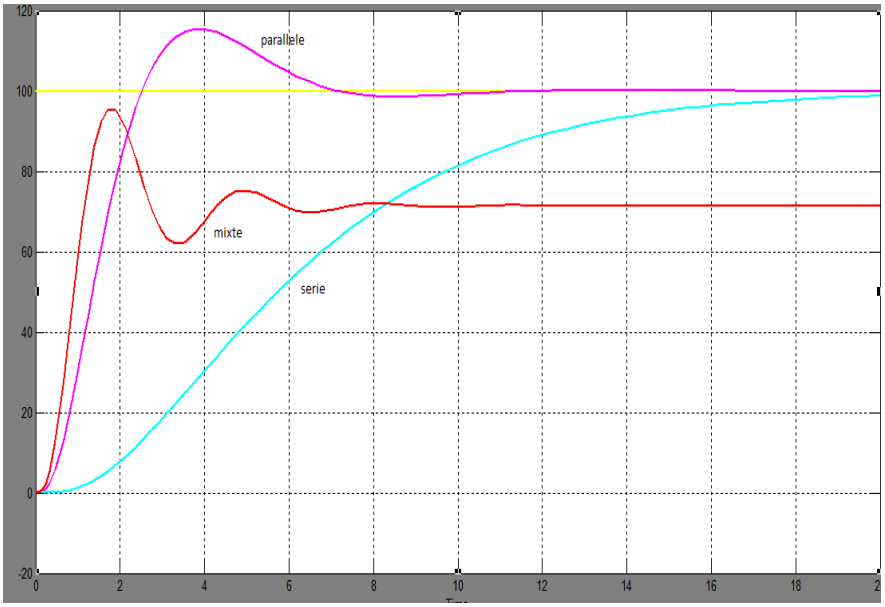


**Figure II-16 :** La réponse du système de 3eme ordre commandé par PID

**Remarque** : on observe que la meilleure réponse est la réponse de la structure parallèle.

On a une autre réponse d'un système avec tous les autres régulateurs précédents :

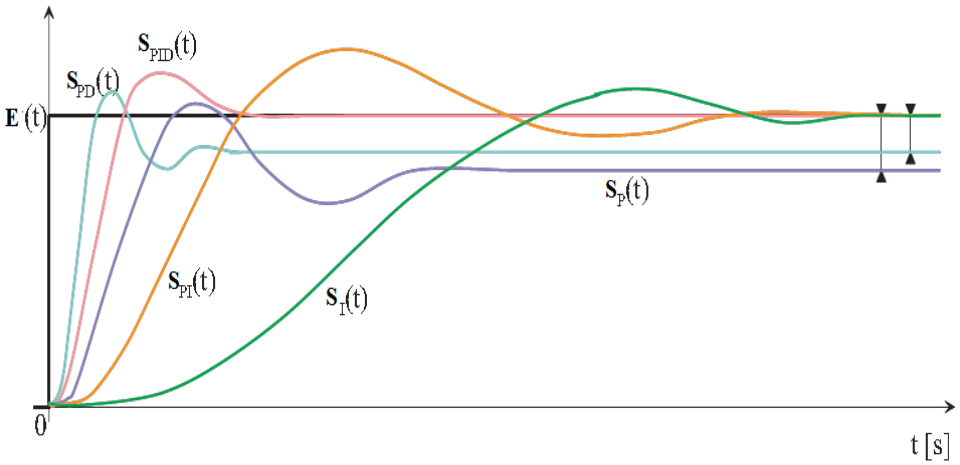


**Figure II-17 :** Réponse d'un système avec tous les autres régulateurs

**Synthèse sur les actions PID**

On résume les avantages et les limitations des actions de base des régulateurs PID :

| action | avantages | désavantages |
|---|---|---|
| P | Action instantanée | Ne permet pas d'annuler une erreur statique mais permet de la réduire |
| I | Annule l'erreur statique | Action lente<br>Ralentit le système (effet déstabilisant) |
| D | Action très dynamique<br>Améliore la stabilité<br>Détruit la rapidité | Sensibilité aux bruits<br>Forte sollicitation de l'organe de commande |

**Tableau1 :** Synthèse sur les actions PID

| Augmentation de | Stabilité | Précision | Rapidité |
|---|---|---|---|
| $kp$ | Diminue | Augmente | Augmente |
| $Ti$ | Augmente | Pas d'influence | Diminue |
| $Td$ | Augmente | Pas d'influence | Diminue |

**Tableau.2 :** Tableau d'influence de PID

**Réglage**

Un critère de réglage d'une boucle de régulation doit permettre de répondre au plus grand nombre de contraintes exigées par le cahier des charges du procédé à réguler. Les besoins en régulation ou asservissement étant tris variés, de nombreuses stratégies de réglage d'une boucle sont possibles. Les exigences du cahier des charges sont décrites soit dans le domaine temporel, soit dans le domaine fréquentiel. Le critère de réglage est alors fixé à partir soit de la forme de la réponse temporelle souhaitée pour un type d'excitation à l'entrée (par exemple la consigne est un échelon de position), soit à partir des marges de stabilité (marges de gain et de phase. Facteur de résonance). Le critère précision est, bien entendu, intrinsèquement lié à celui du rég1age. Le bon réglage est celui qui répondra au meilleur compromis global du cahier des charges.

Réglage par critère temporel

Il s'agit d'obtenir, en chaîne fermée, une réponse temporelle bien définie pour une excitation d'entrée imposée. Généralement on désire, pour une variation de la consigne en échelon de position, soit une réponse du premier ordre, soit une réponse du deuxième ordre apériodique ou périodique amortie. Cela correspond en pratique à écrire un modèle pour la fonction de transfert en chaîne fermée.

L'avantage essentiel d'imposer une certaine fonction de transfert en chaîne fermée est de garantir un degré de stabilité mais aussi un bon compromis entre précision et rapidité.

Attention cependant ! Les fonctions de transfert perturbatrices n'ayant pas été prises en compte lors de l'élaboration du correcteur, leurs influences sur la forme de la Courbe de réponse ne sont pas connues.

## II.4. Conclusion

Dans ce chapitre nous avons donné une idée générale sur les systèmes asservis par les régulateurs PID. L'action associée au régulateur PID permet une régulation optimale en associant les avantages de chaque action : la composante "P" réagit à l'apparition d'un écart de réglage, la composante "I" élimine l'erreur statique et la composante "D" diminue le dépassement. Dans un régulateur PID, il existe plusieurs façons d'associer les paramètres P, I et D. En effet, le régulateur PID peut avoir une structure série, parallèle ou mixte. Il existe de nombreuses méthodes utilisées pour déterminer les paramètres du régulateur PID.

# Chapitre III :

# Matérielle et logicielle

# Chapitre III. Matérielle et logicielle

## III.1. Introduction

Dans ce chapitre, nous expliquons la partie matérielle,les composants utiliséspour contrôler le température, et leurs tâches(arduinouno, le relais,capteur,etc….).

Enfin on va présenter le langage Arduino, son vocabulaire ainsi la structuration d'un programme écrit en Arduino.

## III.2. Partie matérielle

### III.2.1.La carte ArduinoUno

C’est la carte idéale pour découvrir l’environnement ARDUINO. Elle permet à tout débutant de se lancer dans tous ses premiers petits projets. Comme c’est la carte la plus utilisée, il est très facile de se référer aux tutoriels très nombreux sur le net et ainsi de ne pas rester seul dans son exploration.

Sa simplicité devient par contre un utilisateur lorsqu’il s’agit de multiplier les périphériques, de manipuler des algorithmes lourds ou d’interagie avec les OS Androïde pour les quels d’autres cartes ARDUINO sont plus adaptées.

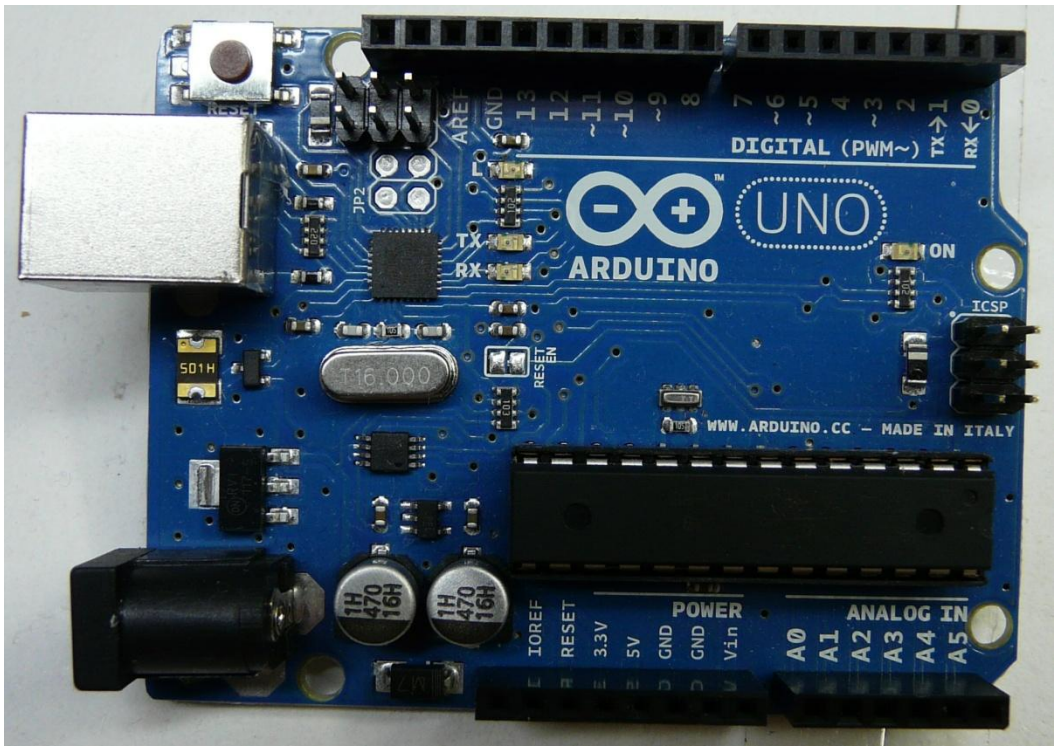


**Figure III-1 :** La carte Arduino uno

### III.2.1.1. Les composants La carte Arduino UNO

La carte Arduino Uno est une carte à microcontrôleur basée sur l'ATmega328 Elle dispose :

- de 14 broches numériques d'entrées/sorties (dont 6 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée)),
- de 6 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques),
- d'un quartz 16Mhz,
- d'une connexion USB,
- d'un connecteur d'alimentation jack,
- d'un connecteur ICSP (programmation "in-circuit"),
- et d'un bouton de réinitialisation (reset).

### III.2.1.2. Synthèse des caractéristiques

| | |
|---|---|
| Microcontrôleur | ATmega328P |
| Tension de fonctionnement | 5V |
| Tension d'entrée (recommandé) | 7-12V |
| Tension d'entrée (limite) | 6-20V |
| E / S numériques Pins | 14 (dont 6 fournissent la sortie PWM) |
| PWM numérique E / S Pins | 6 |
| Pins d'entrée analogique | 6 |
| DC Courant par I O Pin / | 20 mA |
| Courant DC pour 3.3V Pin | 50 mA |
| Mémoire flash | 32 KB (ATmega328P) dont 0,5 KB utilisé par bootloader |
| SRAM | 2 KB (ATmega328P) |
| EEPROM | 1 KB (ATmega328P) |
| Vitesse de l'horloge | 16 MHz |
| Longueur | 68,6 mm |
| Largeur | 53,4 mm |
| Poids | 25 g |

**Tableau III.1 :** caractéristique techniques de la carte Arduinouno

### III.2.1.3. Détails techniques

**Le Microcontrôleur ATMega328P :**

Un microcontrôleur ATMega328P est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit au temps des pionniers de l'électronique. Aujourd'hui, en soudant un grand nombre de composants encombrants ; tels que les transistors; les résistances et les condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C. la figure III.2 montre un microcontrôleur ATmega 328P, qu'on trouve sur la carte Arduino UNO.[3]

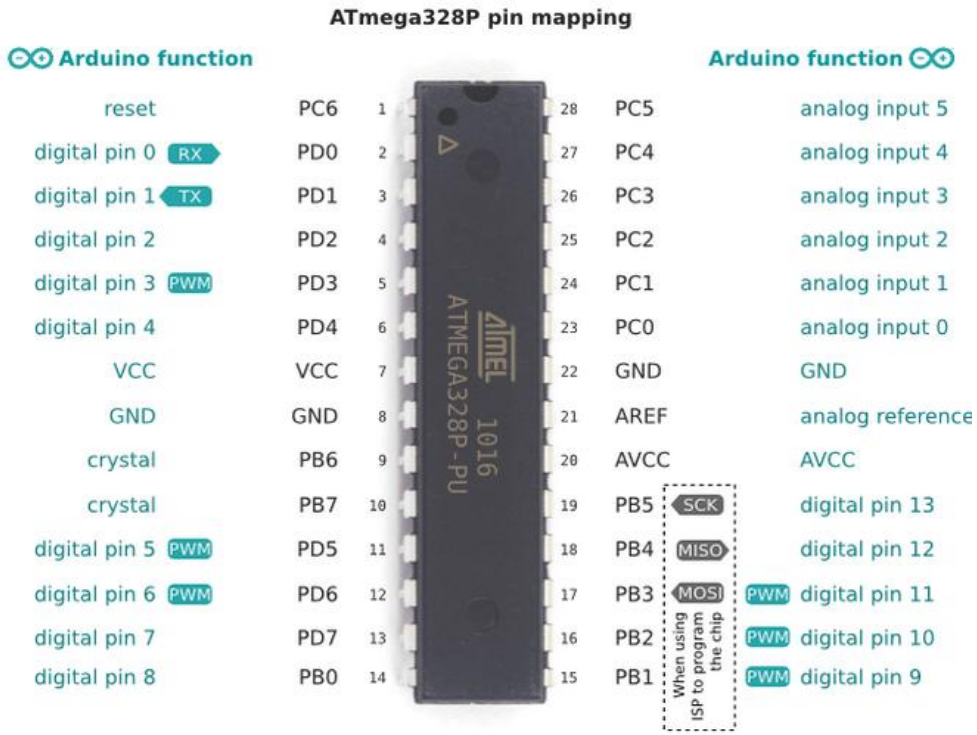


**Figure III-2 :** Microcontrôleur ATMega328P

Le microcontrôleur ATMega328 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement sur :

- **La mémoire Flash :** C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible mémoire programme de 32Ko (dont boot loader de 0.5 ko).
- **RAM:** c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 2 ko.
- **EEPROM:** C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme. [2]

### III.2.1.4. Les sources de l'alimentation de la carte

La carte Arduino Uno peut-être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte.

L'alimentation externe (non-USB) peut être soit un adapteur secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles (ou des accus). L'adaptateur secteur peut être connecté en branchant une prise 2.1mm positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées Gnd (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation.

La carte peut fonctionner avec une alimentation externe de 6 à 20 volts. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte. Aussi, la plage idéale recommandée pour alimenter la carte Uno est entre 7V et 12V.Les broches d'alimentation sont les suivantes :

- VIN. La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). Vous pouvez alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- 5V. La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation
    - VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.

- 3V3. Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V). L'intensité maximale disponible sur cette broche est de 50mA
- GND. Broche de masse (ou 0V).

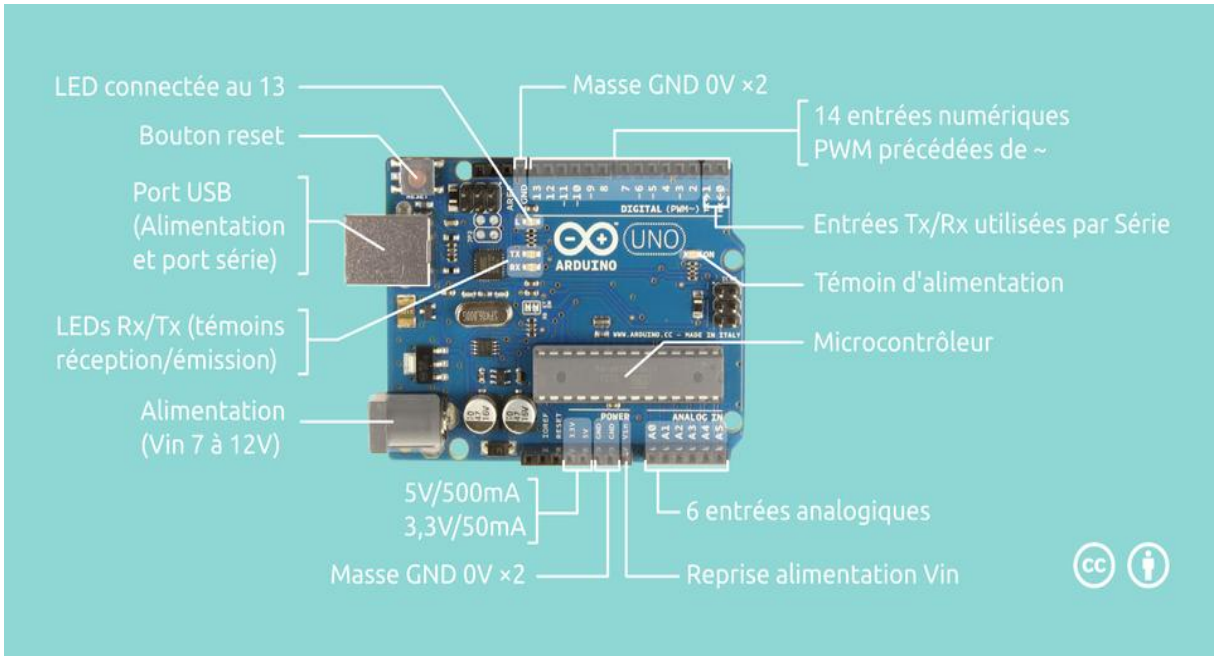


**Figure III-3 :** Constitution de la carte Arduino UNO

### III.2.1.5. Brochage de la carte Arduino uno

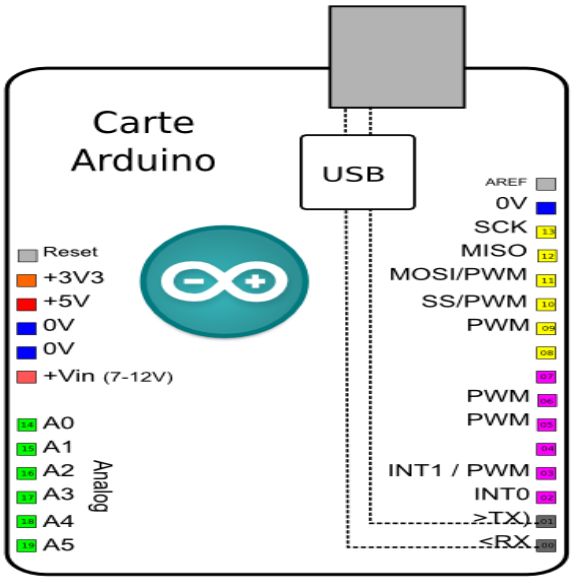


**Figure III-4 :**Brochage de la carte ArduinoUno

### III.2.1.6. Les entrées& sorties

Les entrées / sorties (I/O – input/output) représentent le moyen qu'a la carte Arduino d'interagir avec l'extérieur. Les sorties sont contrôlées par la carte, cela permet au programme du microcontrôleur de déclencher des actions (allumer ou d'éteindre une LED, un ventilateur, un moteur). Les entrées sont lues par le microcontrôleur, ce qui lui permet de connaitre l'état du système auquel il est relie. Il y a deux sortes d'I/O : les I/O numériques, et les I/O analogiques.

- Les entrées / Sorties Numériques :Les entrées / sorties numériques ne peuvent prendre que deux valeurs, la valeur LOW (~ GND, 0 V), et la valeur HIGH (~ 5 V). La valeur d'un port numérique peut donc être codée sur un bit, 0 ou 1, troue ou false.

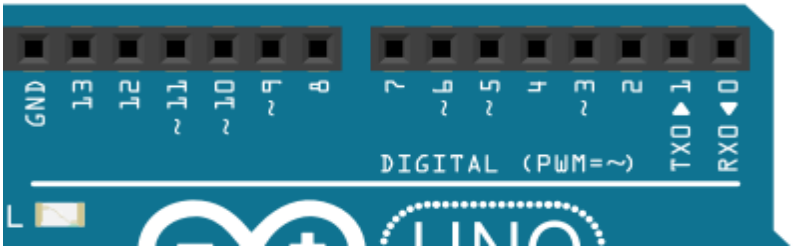


**Figure III-5 :** les entrées/sorites numérique de la carte Arduinouno

La carte Arduino comporte 14 I/O numériques (appelées DIGITAL sur la carte), numérotées de 0a13 (voir la figure IV.3), et appelées D0, D1, D2, … D13. Chacun de ces ports peut-être

Déclare comme étant une entrée ou comme une sortie dans le programme du microcontrôleur.

Les deux premiers ports (D0 et D1) sont réservés a la communication série, il ne faut pas les

Utiliser. Le dernier port, D13, possède un indicateur lumineux, une LED qui s'allume quand le

Port est HIGH, et qui s'éteint quand le port est LOW.

Le port GND est la masse de la carte (0 V).

- Les sorties Numériques : Chacun des 14 ports numériques de la carte peuvent être utilises en sortie. Si un port est déclare comme une sortie, le microcontrôleur contrôle la valeur de ce port. Attention, le courant que peut délivrer un port digital en sortie est limite a 40 mA : en demander plus peut endommager la carte ! Ce genre de situation peut arriver si un port, déclare comme une sortie, est directement relie a la masse (port GND) avec une résistance très faible (un fil), et que le programme bascule la sortie en HIGH (5 V). L'inverse est également dangereux (une sortie numérique reliée au port 5 V et basculée sur la valeur LOW).Les sorties numériques ne peuvent pas fournir une grande puissance électrique (40 mA max sur 5 V). On les utilise pour échanger des

informations (par exemple les ports D0 et D1 servent a la communication série avec l'ordinateur), ou pour déclencher des actions : par exemple allumer une LED.

- Les entrées numériques : Chacun des 14 ports numériques de la carte peuvent être utilises en entrée. Si un port est déclaré comme entrée, l'état du port sera lu par Arduino (HIGH ou LOW), et cette valeur pourra être utilisée dans le programme pour déclencher telle ou telle action. Déclare comme une entrée, un port numérique sera considère comme HIGH ou LOW selon la valeur de la tension mesurée par la carte. En gros, les tensions inferieures à 1 V seront lues comme LOW, les tensions supérieures à 4 V seront lues comme HIGH. Il faut éviter les tensions intermédiaires, qui risquent de donner un résultat indéterminé. Attention, une tension supérieure à 5.5 V peut détruire la carte Arduino.

- Les entrées analogiques : Une entrée analogique est une sorte de voltmètre : la carte lit la tension qui est appliquée sur le port. Cependant le microcontrôleur ne travaille qu'avec des chiffres : il faut donc transformer la tension appliquée en sa valeur numérique. C'est le travail du convertisseur analogue/digital, dit ≪ CAD ≫.

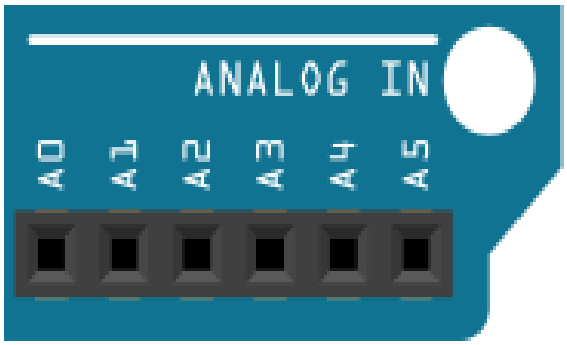


**Figure III-6 :** Les entrées analogiques de la carte Arduino uno

Le CAD de la carte Arduino travaille sur 10 bits : il accepte en entrée une tension comprise entre0 V et Vref une tension de référence, et fournit au microcontrôleur un chiffre entier compris entre0 et 1023 (= 210 - 1). Une tension inferieure à 0 V est lue comme 0, une tension supérieure à Vrefest lue comme 1023, une tension intermédiaire est

lue comme un entier entre 0 et 1023, avec une relation linéaire. La tension Vref est 5 V par défaut, mais cette valeur peut être changée dans le programme.

- Les sorties analogiques : La carte Arduino ne possède pas de vraie sortie analogique, capable de produire une tension d'une valeur arbitraire choisie par l'utilisateur. Certains ports numériques peuvent cependant servir de sortie analogique en utilisant la technique de PWM (Pulse Width Modulation ou bien ' Modulation de largeur d'impulsion') : il s'agit des ports 3, 5, 6, 9, 10 et 11 (signales par un ~ sur la carte). Ces ports peuvent simuler une
- tension entre 0 et 5 V en basculant rapidement entre leur état LOW (O V) et HIGH (5 V).
- La valeur moyenne de la tension est alors 2.5 V si le port passe autant de temps dans un
- état que dans l'autre, mais en changeant ce rapport, la valeur moyenne de la tension peut
- être contrôlée de 0 à 5V.

- La carte Arduino est capable de faire varier la valeur moyenne de ces ports avec une sensibilité de 8 bits : on fournit un chiffre entier compris entre 0 et 255 (= 2^8 - 1), et le port délivre une tension moyenne entre 0 et 5 V (0 = 0 V, 255 = 5 V).

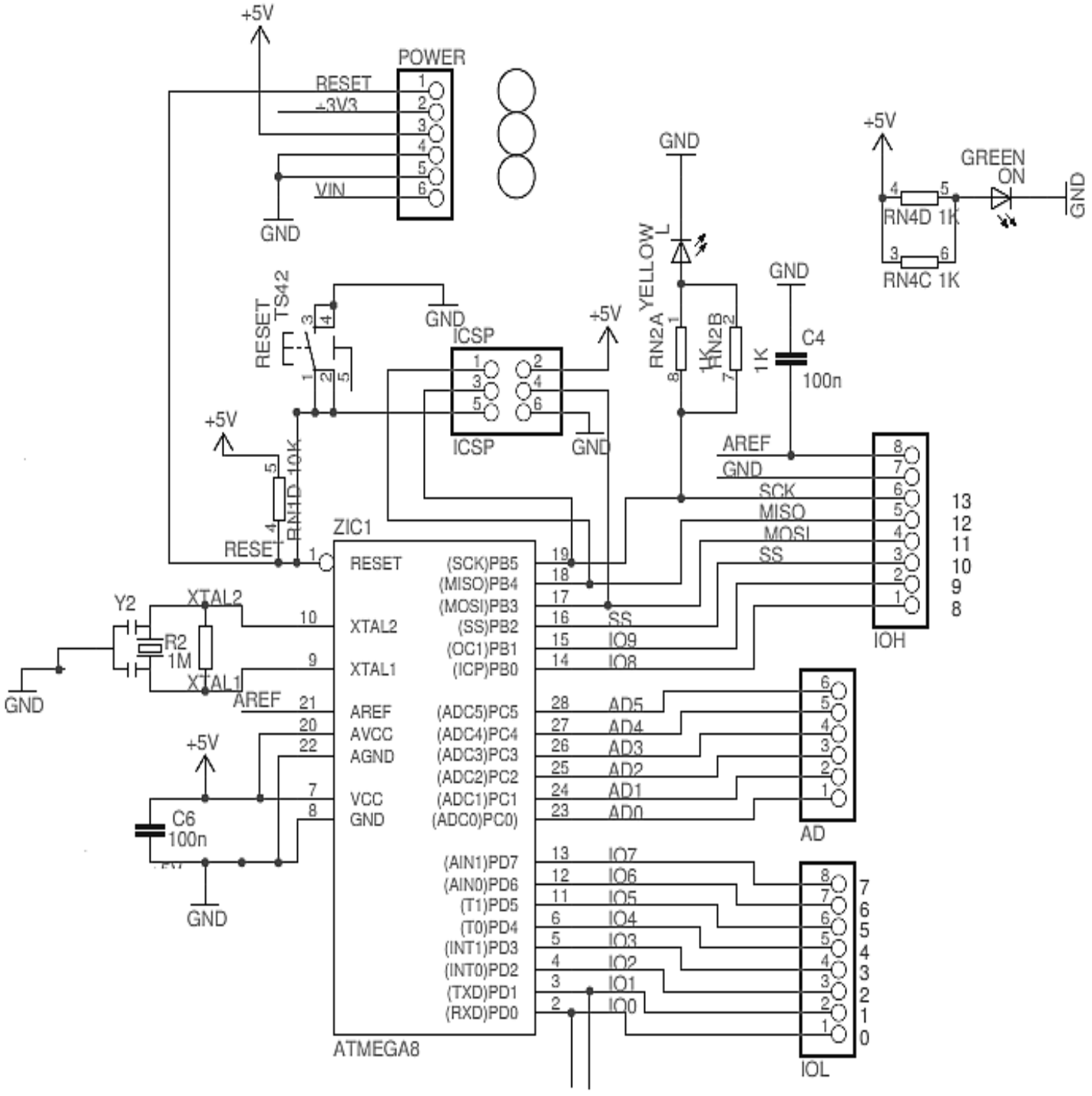


**Figure III-7 : Schéma détaille du carte arduino**

### III.2.1.7. Communications

La carte Arduino Uno dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs. L'ATmega 328 dispose d'une UART (Universal Asynchronous ReceiverTransmitter ou émetteur-récepteur asynchrone universel en français) pour communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX). Un circuit intégré ATmega8U2 sur la carte assure la connexion entre cette communication série vers le port USB de l'ordinateur et apparaît comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'ATmega8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est nécessaire.

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LED RX et TX sur la carte clignote lorsque les données sont transmises via le circuit intégré USB-vers-série et la connexion USB vers l'ordinateur.

Une librairie Série Logicielle permet également la communication série (limitée cependant) sur n'importe quelle broche numérique de la carte UNO.

L'ATmega 328 supporte également la communication par protocole I2C (ou interface TWI (TwoWire Interface - Interface "2 fils") et SPI :

- Le logiciel Arduino inclut la librairie Wire qui simplifie l'utilisation du bus I2C. Voir la documentation pour les détails
- Pour utiliser la communication SPI (Interface Série Périphérique), la librairie pour communication SPI est disponible.

### III.2.2.Le Capteur

Un capteur est un dispositif transformant l'état d'une grandeur physique observée en une grandeur utilisable, telle qu'une tension électrique, une hauteur de mercure, une intensité ou la déviation d'une aiguille. On fait souvent (à tort) la confusion entre *capteur* et transducteur : le *capteur* est *au minimum* constitué d'un transducteur.

Le capteur se distingue de l'instrument de mesure par le fait qu'il ne s'agit que d'une simple interface entre un processus physique et une information manipulable. Par opposition, l'instrument de mesure est un appareil autonome se suffisant à lui-même, disposant d'un affichage ou d'un système de stockage des données. Le capteur, lui, en est dépourvu.

Les capteurs sont les éléments de base des systèmes d'acquisition de données. Leur mise en œuvre est du domaine de l'instrumentation.

### III.2.3.Le Relais

Un relais électromécanique est un organe électrique permettant de dissocier la partie puissance de la partie commande : il permet l'ouverture et la fermeture d'un circuit électrique par un second circuit complètement isolé (isolation galvanique) et pouvant avoir des propriétés

## III.3. Programmation en arduino

### III.3.1.Définition d’un programme arduino

Un langage de programmation est un langage permettant à un être humain d'écrire un ensemble d'instructions (code source) qui seront directement converties en langage machine grâce à un compilateur (c'est la compilation). L'exécution d'un programme Arduino s'effectue de manière séquentielle, c'est-à-dire que les instructions sont exécutées les unes à la suite des autres. Voyons plus en détail la structure d'un programme écrit en Arduino

### III.3.2.La structure d'un programme

Un programme Arduino comporte trois parties :

- la partie déclaration des variables (optionnelle)
- la partie initialisation et configuration des entrées/sorties : la fonction setup ()
- la partie principale qui s'exécute en boucle : la fonction loop ()

Dans chaque partie d'un programme sont utilisées différentes instructions issues de la syntaxe du langage Arduino la figure( II.17)dans la page suivante illustre la structure d'un programme arduino .

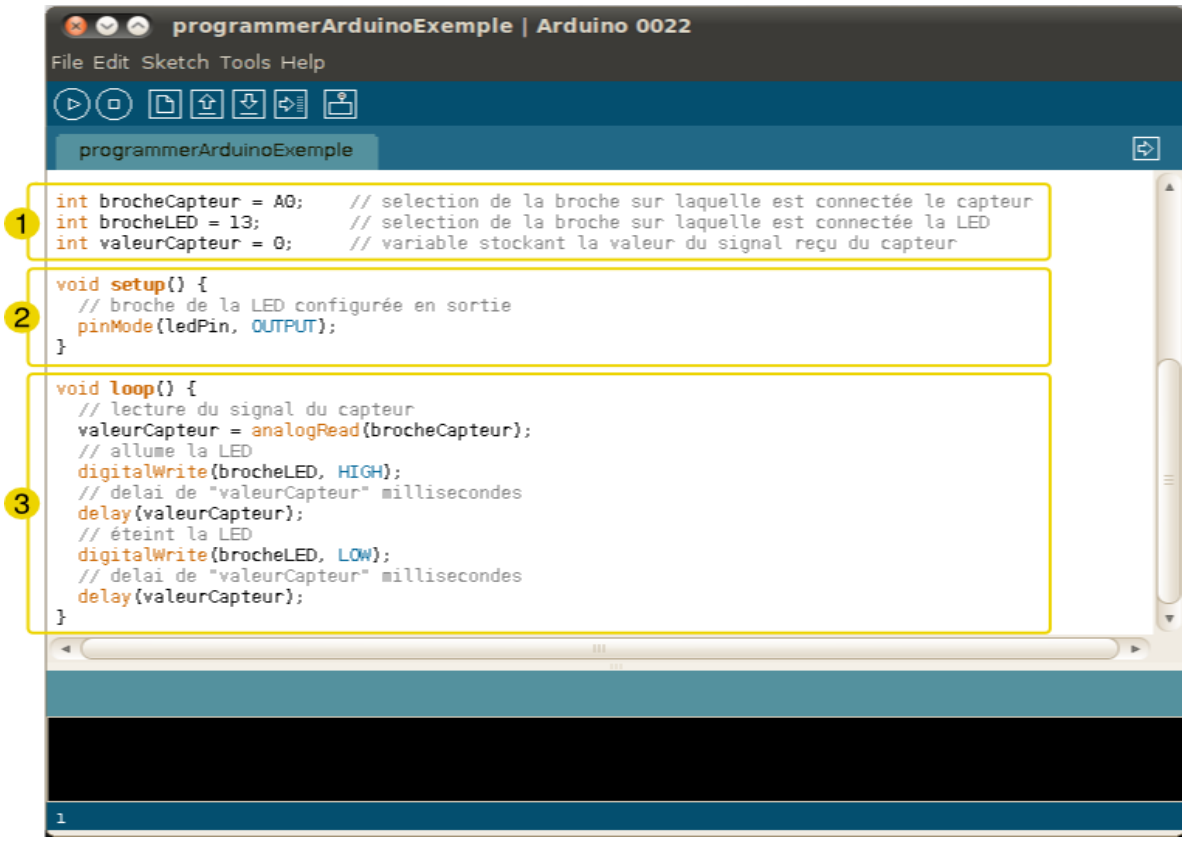


**Figure III-8 :**structure d'un programme arduino

### III.3.3.Coloration syntaxique

Lorsque du code est écrit dans l'interface de programmation, certains mots apparaissent en différentes couleurs qui clarifient le statut des différents éléments :

En orange, apparaissent les mots-clés reconnus par le langage Arduino comme des fonctions existantes. Lorsqu'on sélectionne un mot coloré en orange et qu'on effectue un

clic avec le bouton droit de la souris, l'on a la possibilité de choisir « Find in reference » : cette commande ouvre directement la documentation de la fonction sélectionnée.

En bleu, apparaissent les mots-clés reconnus par le langage Arduino comme des constantes.

En gris, apparaissent les commentaires qui ne seront pas exécutés dans le programme. Il est utile de bien commenter son code pour s'y retrouver facilement ou pour le transmettre à d'autres personnes. L'on peut déclarer un commentaire de deux manières différentes

- dans une ligne de code, tout ce qui se trouve après « // » sera un commentaire.
- l'on peut encadrer des commentaires sur plusieurs lignes entre « /* » et « */ ».

### III.3.4.La syntaxe du langage :

#### III.3.4.1. Ponctuation :

Le code est structuré par une ponctuation stricte :

- toute ligne de code se termine par un point-virgule « ; »
- le contenu d'une fonction est délimité par des accolades « { » et « } »
- les paramètres d'une fonction sont contenus pas des parenthèses « ( » et « ) ».

Une erreur fréquente consiste à oublier un de ces éléments.

#### III.3.4.2. Les variables :

Une variable est un espace réservé dans la mémoire de l'ordinateur. C'est comme un compartiment dont la taille n'est adéquate que pour un seul type d'information. Elle est caractérisée par un nom qui permet d'y accéder facilement.

Il existe différents types de variables identifiés par un mot-clé dont les principaux sont :

- nombres entiers (int)
- nombres à virgule flottante (float)
- texte (String)

- valeurs vrai/faux (boolean).
- Un nombre à décimales, peut se stocker dans une variable de type float. Notez que l'on utilise un point et non une virgule pour les nombres à décimales.

Dans Arduino, il est nécessaire de déclarer les variables pour leurs réserver un espace mémoire adéquat. On déclare une variable en spécifiant son type, son nom puis en lui assignant une valeur initiale (optionnelle).

### III.3.4.3. Les fonctions :

Une fonction (également désignée sous le nom de procédure ou de sous-routine) est un bloc d'instructions que l'on peut appeler à tout endroit du programme. Le langage Arduino est

constitué d'un certain nombre de fonctions par exemple : analogRead(),analogWrite(),digitalRead(),digitalWrite() ou delay()

### III.3.4.4. Les structures de contrôle :

Les structures de contrôle sont des blocs d'instructions qui s'exécutent en fonction du respect d'un certain nombre de conditions.

Il existe quatre types de structure :

**if...else** : exécute un code si certaines conditions sont remplies et éventuellement exécutera un autre code avec sinon.

```
//si la valeur du capteur dépasse le seuil

if(valeurCapteur>seuil){

//appel de la fonction clignote

clignote();}
```

while : exécute un code tant que certaines conditions sont remplies.

**Exemple :**

```
//tant que la valeur du capteur est supérieure à 250

while(valeurCapteur>250){

  //allume la sortie 5

digitalWrite(5,HIGH);

  //en boucle tant que valeurCapteur est supérieure à 250

}

digitalWrite(5,LOW);
```

for : exécute un code pour un certain nombre de fois.

**Exemple :**

```
//pour i de 0 à 255, par pas de 1

for (int i=0; i <= 255; i++){analogWrite(PWMpin, i);

delay(10);

}
```

switch/case : fait un choix entre plusieurs codes parmi une liste de possibilités.

**Exemple :**

```
// fait un choix parmi plusieurs messages reçus
switch (message) {
   case 0: //si le message est "0"
      //allume que la sortie 3
digitalWrite(3,HIGH);
digitalWrite(4,LOW);
digitalWrite(5,LOW);
break;
   case 1: //si le message est "1"
      //allume que la sortie 4
digitalWrite(3,HIGH);
digitalWrite(4,LOW);
digitalWrite(5,LOW);
break;
   case 2: //si le message est "2"
      //allume que la sortie 5
digitalWrite(3,LOW);
digitalWrite(4,LOW);
digitalWrite(5,HIGH);
break;
   }
```

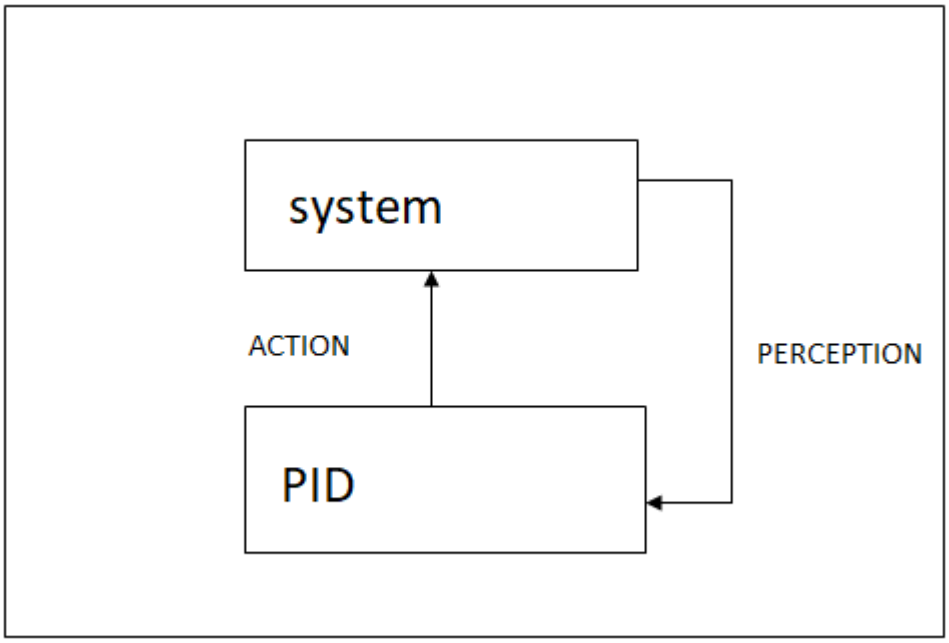


**Figure III-9 :** schéma général du system

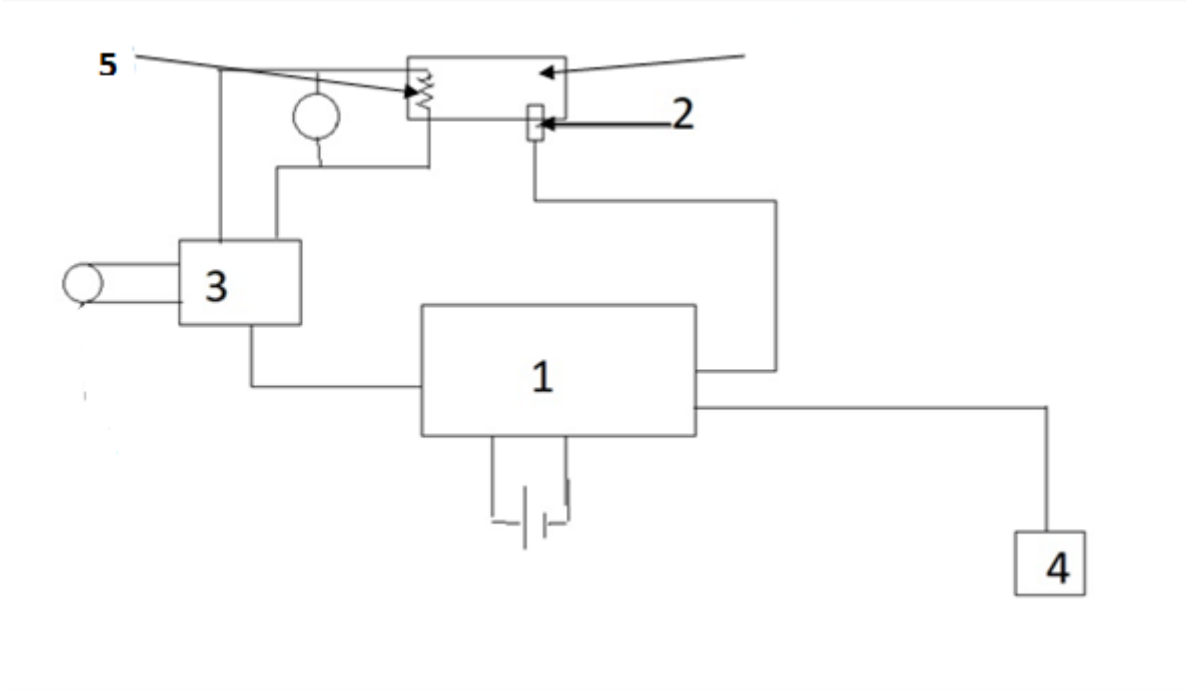


**Figure III-10 :** schéma détaillé du system

1-carte arduino : calcul et commander le relais celons le résultat opte nie du correcteur

Envoie message (0/1) si message 0 le relais désactivée actionneur, si message 1 le relais activée actionneur.

2-capteur (thermocouple : capture la température (mesurée la température)

3-le relais : activité et désactivé l'actionneur

4-pc (écran) : affichée la température.

5-resistance

## III.5. Conclusion :

Dans ce chapitre, on a fait une étude approfondit sur les composants utilisés dans notre projet (arduino uno, le relais ,capteur, etc….) et on a essayé de donner des définitions sur les deux programmes utilisés dans le projet (l'arduino et PID) et présenter les parties plus important dans les deux programme, dans le chapitre suivant on va faire une réalisation du projet avec les composants cités fera dans le chapitre suivant.

# Chapitre IV :
# L’implémentation

# Chapitre IV. L'implémentation

## IV.1. Introduction

Dans ce chapitre, nous présenterons le résultat final de nos travaux.

- Le tableau suivant indique le changement de température d'une plaque métallique que nous avons introduite dans un système. Nous atteindrons le degré que nous avons identifié tout en minimisant le taux d'erreur.

| temps | température de la plaque | Température de référence | erreur |
|---|---|---|---|
| 1 | 35 | 50 | 15 |
| 2 | 35 | 50 | 15 |
| 3 | 35 | 50 | 15 |
| 4 | 35 | 50 | 15 |
| 5 | 35 | 50 | 15 |
| 6 | 35 | 50 | 15 |
| 7 | 35 | 50 | 15 |
| 8 | 35 | 50 | 15 |
| 9 | 35 | 50 | 15 |
| 10 | 35,5 | 50 | 14,5 |
| 11 | 35,5 | 50 | 14,5 |
| 12 | 35,5 | 50 | 14,5 |
| 13 | 35,5 | 50 | 14,5 |
| 14 | 35,5 | 50 | 14,5 |
| 15 | 35,5 | 50 | 14,5 |
| 16 | 36 | 50 | 14 |
| 17 | 37 | 50 | 13 |
| 18 | 37 | 50 | 13 |
| 19 | 37 | 50 | 13 |
| 20 | 37,5 | 50 | 12,5 |
| 21 | 37,5 | 50 | 12,5 |
| 22 | 38 | 50 | 12 |
| 23 | 38 | 50 | 12 |
| 24 | 38,5 | 50 | 11,5 |

| | | | |
|---|---|---|---|
| 25 | 39 | 50 | 11 |
| 26 | 39,5 | 50 | 10,5 |
| 27 | 39,5 | 50 | 10,5 |
| 28 | 39,5 | 50 | 10,5 |
| 29 | 40 | 50 | 10 |
| 30 | 40,5 | 50 | 9,5 |
| 31 | 41 | 50 | 9 |
| 32 | 41,5 | 50 | 8,5 |
| 33 | 42 | 50 | 8 |
| 34 | 42,5 | 50 | 7,5 |
| 35 | 43 | 50 | 7 |
| 36 | 43,5 | 50 | 6,5 |
| 37 | 44 | 50 | 6 |
| 38 | 44,5 | 50 | 5,5 |
| 39 | 45 | 50 | 5 |
| 40 | 46 | 50 | 4 |
| 41 | 47 | 50 | 3 |
| 42 | 47,5 | 50 | 2,5 |
| 43 | 47,5 | 50 | 2,5 |
| 44 | 48 | 50 | 2 |
| 45 | 48,5 | 50 | 1,5 |
| 46 | 49 | 50 | 1 |
| 47 | 50 | 50 | 0 |
| 48 | 50,5 | 50 | -0,5 |
| 49 | 50 | 50 | 0 |
| 50 | 50 | 50 | 0 |
| 51 | 49,5 | 50 | 0,5 |
| 52 | 50 | 50 | 0 |
| 53 | 49,5 | 50 | 0,5 |
| 54 | 50 | 50 | 0 |
| 55 | 50 | 50 | 0 |
| 56 | 50 | 50 | 0 |
| 57 | 49,5 | 50 | 0,5 |
| 58 | 50 | 50 | 0 |
| 59 | 50 | 50 | 0 |

**Table IV-1** : Résultat final de travaille

Température de la plaque : Température de la plaque métallique

Température de référence : Température Sélectionné

Erreur : Ratio d'erreur

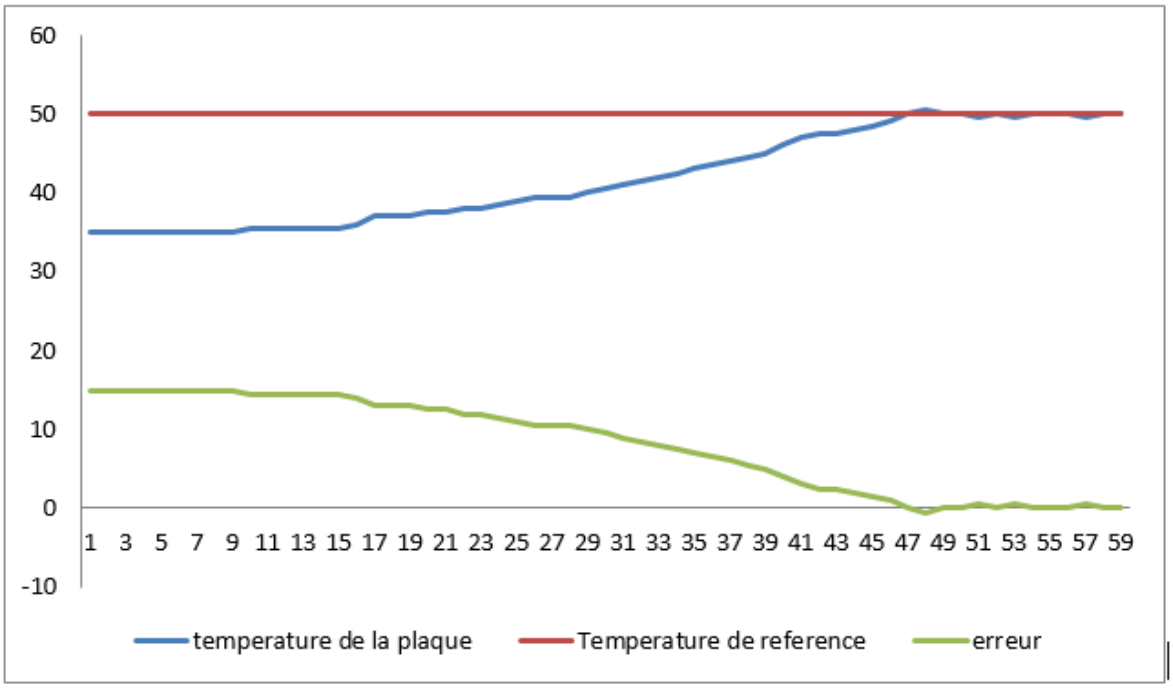


**Figure IV-1 :** Courbe de changement températureet l’erreur

Le courbe dans la figure (IV.1) présente le changement de la température de la plaque(en bleu) dans un intervalle de temps bien précisé (60 secondes) par rapport la température référence (50 dégrées ) présenté en rouge , ainsi la variation du valeur de l’erreur du système durant la période de régulation .On remarque que a chaque fois la température de

la plaque se rapproche vers la température de référence 50 degrés l’erreur s’annule et c’est grâce au correcteur utilisé : PID

On constate que le correcteur PID est un correcteur édial pour les application de régulation de température .Il nous a donné des bon résultats avec la carte arduino uno :

- Rapidité du système
- Stabilité du système
- L’erreur obtenue de le test est correcte

## IV.2. Conclusion

Dans ce chapitre, nous avons présenté le résultat de nos travaux, qui montre que le contrôleur a contrôleur PID la température tout en minimisant le taux d’erreur.

## Conclusion Générale

La réalisation des contrôleurs automatiques ont été le cœur de plusieurs projets de recherche lors de ces dernières années. Ce travail a un objectif de réaliser un système intelligent pour améliorer la contrôle de température en utilisant le PID a fin d'obtenir un système automatique fiable.

Alors au début de notre thème, nous avons essayé de donner un aperçu sur l'asservissement .Ensuite, dans le deuxième chapitre on a essayé de présenter le terme PID. Enfin, le dernier chapitre est intéressé aux la description de système réalisé en décrivant le raisonnement du programme

En perspectives, nous pouvons dire que ce travail n'est qu'une simple application dans le domaine de contrôleurs, il peut être plus autonome, plus pratique, Parmi les perspectives ouvertes à ce projet :

- Améliorer l'interface de commande de l'application
- Améliorer la commande de l'application en ajoutant différent type de capteur, de logiciels, et de matériel informatique

[1] F.Alouani.2006. Magister en électronique : option control, commande par logique floue appliquée aux pendules inversés et au simulateur de vol d'hélicoptère : Simulation et expérimentation, université de Mohamed Boudiaf de M'sila,

[2]V.Boitier. 2005. Asservissement linéaires continus. Université Paule Sabatier Toulouse

[3]M. Sahnoun « Conception et Simulation d'une Commande à Retour d'Effort pour Fauteuil Roulant Electrique » Thèse Doctorat, Université de Metz 2007.

[4] R. Grasse « Aide à la Navigation pour les Personnes Handicapées : Reconnaissance deTtrajets » Thèse Doctorat, Université de Metz 2007